University of Zagreb

Faculty of Mechanical Engineering and Naval Architecture

MASTER THESIS

Ivan Zenzerović

Zagreb, 2012

University of Zagreb

Faculty of Mechanical Engineering and Naval Architecture

Numerical Modelling of Dynamics of Multibody Systems in Lie-group Setting

Mentor:

Prof. dr. sc. Zdravko Terze

Student:

Ivan Zenzerović

Zagreb, 2012

Izjavljujem da sam ovaj rad izradio samostalno koristeći stečena znanja tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru prof. dr. sc. Zdravku Terzeu na presenom znanju i izvrsnom vodstvu tijekom izrade ovog rada te na pokazanom strpljenju i udijeljenim savjetima bez kojih ovaj rad ne bi bilo moguće izraditi.

Zahvale upućujem i Dariu Zlataru na ukazanoj pomoći prilikom izrade ovog rada.

Zahvaljujem se svojim roditeljima Stanku i Lučani te bratu Robertu na omogućenom školovanju i pruženoj podršci tijekom cijelog studija.

Posebno se zahvaljujem i svojoj zaručnici Željki Budiselić na pruženoj potpori i strpljenju tijekom izrade ovog diplomskog rada.

Bez vas ništa od ovoga ne bi bilo moguće.

Ivan Zenzerović

Contents

List of Figures

List of Tables

List of Symbols

Symbol	Unit	Description
n	-	Number of system degrees of freedom
N	-	Number of bodies/particles in a system
K	-	Number of constraints
m_i	kg	Body i mass
$\mathbf{J}_i$	kgm^2	Body i inertia matrix
J_k^i	kgm^2	Body i mass moment of inertia about the k-axis
$\mathbf{F}_i$	N	Body i external force vector
$\mathbf{L}_i$	Nm	Body i external torque vector
$\mathbf{Q}$	N, Nm	System generalized force vector
$\mathbf{Q}_{nl}$	N, Nm	System non-linear velocity forces vector
$\mathbf{Q}_{ext}$	N, Nm	System external forces vector
$\mathbf{M}$	kg, kgm^2	System inertia matrix
$\boldsymbol{\lambda}$		Lagrange multipliers vector
$\boldsymbol{\Phi}_x$		System constraint matrix
$\boldsymbol{\xi}$		System acceleration level constraint right-hand side vector
$\dot{\boldsymbol{\Phi}}$		System velocity level constraint matrix
$\boldsymbol{\Phi}$		System displacement level constraint matrix
T	J	Kinetic energy
V	J	Potential energy
L	J	Lagrangian
$\mathbb{R}^n$		n-dimensional Euclidean space
$\mathbb{Q}^n$, $\mathbb{M}^n$		n-dimensional manifold
$SO\,(3)$		The rotations Lie group
$so\,(3)$		The rotations Lie algebra
e		Identity element of a group

$\mathbf{X}_i^p$	m	Vector of the joint p position on body i (in body-fixed CS)
$\mathbf{x}_i$	m	Body i global position vector
$\dot{\mathbf{x}}_i$	m	Body i translational velocity
G_i		Body i configuration space
G		System configuration space
S		System state-space
$\mathbf{I}$, $\mathbf{I}^{p\times r}$		Unitary matrix (of dimension $p \times r$)
$\mathbf{0}$, $\mathbf{0}^{p\times r}$		Zero matrix (of dimension $p \times r$)
$\mathbf{e}^k$		Unit vector along the k axis
$\mathbf{n}$, $\vec{n}$		A vector
$\widetilde{\mathbf{n}}$		A skew-symmetric matrix (formed out of a vector)
θ_i	rad	i-th Euler rotation angle
ϕ	rad	Body rotation angle (of the Euler theorem)
$\mathbf{R}_i$		Body i rotation matrix
$\dot{\mathbf{R}}_i$		Body i rotation matrix time derivative
$\boldsymbol{\omega}_i$, $\boldsymbol{\Omega}_i$	rad/s	Body i angular velocity vector
$\dot{\boldsymbol{\omega}}_i$, $\dot{\boldsymbol{\Omega}}_i$	rad/s^2	Body i angular acceleration vector
t	s	Time
h	s	Integration time step length
T_{sim}, T_{end}	s	Simulation end time
T_{exp}, T_{cay}	s	Computation time of the solution implementation

Abstract

A Lie group integration method, proposed in [12] and [11], is presented in the thesis and applied to the case study problem. The way dynamics of multibody system is mathematically modelled is presented and the governing equations, in the form of a DAE system of index 1 and 3, are given. Manifolds are introduced and applied to the dynamics of MBS. Also, the mathematical background of manifolds, groups, vector fields and spaces is presented in the thesis. The special orthogonal group $SO(3)$ is presented in detail and the concept of Lie groups defined and the group properties described. A Lie group integration method proposed in [12] and [11] is described and its mathematical framework presented and discussed in more details. Two different MBS integration algorithms based on Lie group setting, that include the Lie group Euler and MK method respectively, are described. Also, two mapping functions, that map elements from the Lie algebra to the Lie group, are presented. The Lie group integration methods described are demonstrated on the case study problem of a satellite with a manipulator system on it. Joints between bodies are described and constraint equations formulated at the displacement, velocity and acceleration level are derived. The system bodies are identified and inertial and geometrical parameters given. Rheonomic constraints, giving the imposed motions in the system, are also formulated. Finally, the system is solved using both Lie group integration methods. The results are compared with the *ADAMS* solution. The Lie group methods solutions are obtained from the *MATLAB* algorithms implementation. Different motion cases, for different motions and operations of the satellite and its manipulator, are analysed. The results of the analyses are post-processed, presented and conclusions given.

Keywords: *MBS Mathematical Modelling, Manifolds, DAE System, Lie group, Multybody System Dynamics, Lie group Euler Method, Munthe-Kaas Integration Method, Special Orthogonal Group $SO(3)$, Joints Constraints Formulation, Constraint Violation Stabilisation*

Extended abstract (Croatian)

Sažetak

Integracijska metoda za integriranje dinamike konstrukcijskih sustava na Lievoj grupi, predložena u [12] i [11], opisana je i primijenena za rješavanje gibanja satelita s manipulatorom. U radu je detaljno opisan matematički aparat koji omogućava primjenu linearnih integracijskih metoda na nelinearnim prostorima koji posjeduju posebna svojstva. S obzirom da metoda svoje prednosti pokazuje na prostornim rotacijama tijela, poseban je na naglasak stavljen na grupu specijalnih ortogonalnih matrica $SO\,(3)$, koja je također detaljno opisana. Nakon opisa Lie group Euler i Munthe-Kaas integracijskih metoda, iste su primijenene za rješavanje gibanja satelita i njegovog manipulatora. Analizirana su tri različita slučaja gibanja satelita i manipulatora: za različite operacije te slučaj kvara manipulatora. Integracijski algoritmi su implementirani u *MATLAB*-u, a verifikacija rezultata je provedena uz pomoć rezultata iz *ADAMS*-a. Iz dobivenih rezultata izvučeni su zaključci o korištenim integracijskim metodama čija je osnovna prednost da se prilikom integracije sustava ne javljaju kinematički singulariteti. Singulariteti bi se neizbježno pojavljivali u slučaju velikih domena rotacijskih gibanja da je dinamika sustava modelirana na vektorskom prostoru primjenom neke od lokalnih parametrizacija prostornih (3D) rotacija (npr. Eulerovi kutevi).

Ključne riječi: *Modeliranje dinamike konstrukcijskih sustava, mnogostrukosti, sustav algebarsko-diferencijalnih jednadžbi, Lieva grupa, dinamika konstrukcijskih sustava, Lie group Euler metoda, Munthe-Kaas integracijska metoda, specijalna ortogonalna grupa $SO\,(3)$, jednadžbe kinematičkih ograničenja, stabilizacija povrede ograničenja*

Prošireni sažetak

U prvom poglavlju rada dane su osnove modeliranja mehanike i matematičke osnove dinamike konstrukcijskih sustava. Opisani su stupnjevi slobode gibanja, kinematička ograničenja

i način određivanja broja stupnjeva slobode gibanja. Prije samog uvođenja pojma stupnjeva slobode gibanja, ukratko je opisan prostor i uvedena je metrika prostora. Time su opisane matematičke osnove za opisivanje gibanja i konfiguracije sustava u prostoru.

Broj stupnjeva slobode gibanja sustava je broj nezvisnih koordinata (parametara, brojeva) koji u potpunosti opisuju trenutnu konfiguraciju (položaj) sustava. Za sustav čestica u prostoru se broj stupnjeva slobode računa prema izrazu

$$n = 3N - K,$$

gdje je N broj čestica u sustavu, a K broj kinematičkih veza među česticama (broj oduzetih stupnjeva slobode, odnosno broj algebarskih jednadžbi kojim se modeliraju linearno nezavisna kinematička ograničenja). Za sustav tijela u prostoru broj stupnjeva slobode se određuje prema izrazu

$$n = 6N - K.$$

Nakon definicije stupnjeva slobode uveden je pojam konfiguracijskog prostora i konfiguracijske mnogostrukosti. Konfiguracijski prostor je linearni vektorski prostor i, u slučaju sustava tijela bez nametnutih veza među tijelima, se podudara s konfiguracijskom mnogostrukosti. Kada su u sustavu prisutne kinematičke veze sustav više ne može zauzeti proizvoljni položaj u konfiguracijskom prostoru, već se njegova rješenja kreću na konfiguracijskoj mnogostrukosti. Na konfiguracijsku mnogostrukost može se gledati kao na nelinearnu hiperplohu u konfiguracijskom prostoru sustava koja je određena s jednadžbama ograničenja sustava. Za sustav tijela u prostoru linearni konfiguracijski prostor je Euklidski $\mathbb{E}^{6N}$ prostor, a konfiguracijska mnogostrukost za sustav s ograničenjima je nelinearna hiperploha označena s $\mathbb{Q}^{6N-K}$. Hiperploha je nelinearna jer su u općem slučaju i jednadžbe ograničenja nelinearne. Temeljem opisa konfiguracijskog prostora i mnogostrukosti definirane su i poopćene koordinate kao koordinate na mnogostrukosti.

U nastavku poglavlja je pokazano kako jednadžbe dinamike konstrukcijskih sustava (sustava sačinjenog od više međusobno povezanih tijela u prostoru) tvore sustav algebarsko-diferencijalnih jednadžbi. Ovisno o tome jesu li jednadžbe ograničenja izražene na razini pozicija ili ubrzanja, sustav algebrasko-diferencijalnih jednadžbi može biti indeksa 1 ili indeksa 3. Sustav algebarsko-diferencijalnih jednadžbi indeksa 3, koji opisuje dinamiku konstrukcijskog sustava, dan je jednadžbom

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_x^T \\ \mathbf{\Phi}_x & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \boldsymbol{\xi} \end{bmatrix},$$

i pogodan je za rješavanje dinamike sustava integracijskim metodama na Lievim grupama opisanima u nastavku. Metode je moguće primijeniti ukoliko je sustav pravilno konfiguriran u prostoru stanja, modeliranom kao Lieva grupa.

Pored osnova o konstrukcijskim sustavima, u prvom poglavlju su dane i matematičke osnove potrebne za razumijevanje mnogostrukosti, vektorskih polja i vektorskih prostora na mnogostrukostima. Također su opisane i funkcije i krivulje te komutator kao operacija između vektoriskih polja. Na kraju poglavlja, gore navedeni matematički entiteti su primijeneni na dinamiku konstrukcijskih sustava.

Mnogostrukost se može razumijeti kao n-dimenzionalna nelinearna hiperploha koja je lokalno definirana kinematičkim ograničenjima u Euklidskom vektorskom prostoru $\mathbb{E}^p$, čija je dimenzija $p \geq n$. Također, može se reći da je mnogostrukost matematički entitet koji lokalno sliči Euklidskom vektorskom prostoru, što znači da je lokalna topologija mnogostrukosti jednaka topologiji vektorskog prostora. Mnogostrukost se lokalno prikazuje u mapi kojom se ujedno na mnogostrukost lokalno uvodi koordinatni sustav. Kolekcija svih lokalnih mapa na mnogostrukosti se naziva atlasom.

U drugom poglavlju detaljno se opisuju prostorne rotacije tijela i specijalna ortogonalna grupa $SO\,(3)$ zajedno sa svojim svojstvima. Rotacija je opisana kao proces promjene orijentacije tijela, dok je orijentacija stanje tijela. Rotacije se matematički opisuju matricama rotacije $\mathbf{R}$ čija su svojstva navedena u radu. Prije nego je pokazano da matrice rotacije tijela u prostoru čine grupu specijalnih ortogonalnih matrica, definirane su grupe kao matematički entitet sa svojim aksiomima i svojstvima. Specijalna ortogonalna grupa je definirana s

$$SO\,(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} : \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det\left(\mathbf{R}\right) = +1 \right\}.$$

Tom definicijom, specijalna ortogonalna grupa je definirana kao podgrupa opće linearne grupe 3×3 matrica s ograničenjem ortogonalnosti i determinantom jednakom jedinici. Pokazuje se kako je upravo tim ograničenjem u općoj linearnoj grupi matrica definirana mnogostrukost. Kako je $SO\,(3)$ mnogostrukost sa matematičkom strukturom grupe, može se definirati kao Lieva grupa, što omogućava primjenu integracijskih metoda na Lievim grupama u slučaju prostorne rotacije tijela. U poglavlju je objašnjena i Lieva algebra, te je opisana i diferencijalna jednadžba na $SO\,(3)$ grupi čije je rješenje definirano eksponencijalnom mapom. Time je u radu dan potpuni matematički okvir za opis integracijske metode za dinamiku konstrukcijskih sustava na Lievoj grupi. Također, s obzirom da je $SO\,(3)$ mnogostrukost koja može biti lokalno prikazana u prikladnom koordinatnom sustavu vektorskog prostora (npr. parametrizacija Eulerovim kutevima), dan je kratak osvrt na parametrizacije rotacija.

Upravo su različite lokalne parametrizacije mape na $SO(3)$.

U trećem poglavlju je detaljno opisan postupak parametrizacije i integracije na $SO(3)$ grupi. Prilikom integracije u svakoj točki $SO(3)$ grupe se uvodi lokalna parametrizacija uz pomoć inverzne diferencijalne eksponencijalne mape kojom se diferencijalne jednadžbe sa grupe prenose na Lievu algebru koja je linearan prostor. Na algebri se tada provodi integracija pomoću linearnih integracijskih metoda (Eulerova i Runge-Kutta metoda), a rješenje se vraća na grupu pomoću eksponencijalne mape. Na taj način dobiva se nova točka na $SO(3)$ grupi koja predstavlja orijentaciju tijela u trenutnom integracijskom koraku.

Kako su jednadžbe kinematičkih ograničenja formulirane na razini ubrzanja, prilikom integracije dolazi do povreda ograničenja na razini brzina i položaja. Prilikom takve povrede ograničenja rješenje sustava više se ne nalazi na konfiguracijskoj mnogostrukosti već odstupa (eng. *drift*) s nje. Povreda ograničenja u kinematičkim vezama može uzrokovati veće greške u rješenju dinamike sustava, a da bi se to izbjeglo potrebno je rješenje dobiveno integratorom stabilizirati i povratiti ga na konfiguracijsku mnogostrukost. U radu je opisan postupak stabilizacije prilikom kojeg se, u svakom integracijskom koraku, rješava problem najmanjih kvadrata tako da rješenje sustava zadovoljava ograničenja na razini brzina i položaja.

Na kraju trećeg poglavlja dani su i izrazi kojima se računaju različite mape preslikavanja s Lieve algebre na Lievu grupu. Osim eksponencijalne mape, koja se u slučaju $SO(3)$ grupe računa pomoću Rodriguesove jednadžbe, u radu se koristi i Cayleyeva mapa kao još jedna mapa koja povezuje elemente Lieve algebre s elementima Lieve grupe.

U četvrtom poglavlju opisane integracijske metode su korištene za dobivanje rješenja konstrukcijskog sustava. Sustav se sastoji od satelita i manipulatora na njemu koji služi za održavanje satelita i prihvat eksternih objekata. Satelit s manipulatorom se sastoji od četiri tijela: glavnog tijela satelita, temeljnog člana manipultora, nosača klizača i klizača na koji je montiran alat. Kinematičke veze u sustavu se nalaze između tijela tako da je tijelo satelita povezano s temeljnim članom manipulatora putem sfernog zgloba, temeljni član s nosačem klizača putem rotacijskog zgloba, a nosač klizača i klizač su povezani prizmatičnom vezom. Još jedna kinematička veza u sustavu definirana je između tijela satelita i nepomične okoline. Tom se vezom propisuje da jedna točka satelita ostaje nepomična u inercijskom koordinatnom sustavu (koordinatni sustav nepomične okoline).

U radu su, nakon što su tijela opisana i njihova inercijska i geometrijska svojstva određena, izvedene jednadžbe kinematičkih ograničenja za svaku vezu u sustavu. Prikazane su jednadžbe na razini položaja, brzine i ubrzanja, a svaka je matematička formulacija potkrijepljena objašnjenjem i geometrijskom interpretacijom. Također, pojedinim vezama i tijelima

sustava nametnute su reonomne veze, tj. kinematičke veze u čijoj formulaciji je prisutno vrijeme kao parametar. Takve veze smanjuju broj stupnjeva slobode gibanja time što tijelu nameću gibanje koje se mora ostvariti. U sklopu analiziranog sustava prisutne su i takve kinematičke veze. Gibanja su nametnuta na glavno tijelo satelita, na sferni zglob kojim je ograničeno gibanje temeljnog člana manipulatora te, ovisno o slučaju gibanja, na prizmatičnu vezu.

U radu su analizirana tri različita slučaja gibanja satelita s manipulatorom. Za svako gibanje definirana je različita kombinacija nametnutih gibanja i sila koje djeluju na sustav. Analizirani slučajevi gibanja sustava jesu:

Slučaj 1 Na tijelo satelita nametnuta je rotacija s jednim stupnjem slobode (rotacija oko jedne osi) s konstantnom kutnom brzinom $\mathbf{\Omega}_1^{G,1}$. Rotacija $\mathbf{\Omega}_2^{G,1}$ s jednim stupnjem slobode je nametnuta i na temeljni član manipulatora. Time je ostvarena relativna rotacija u sfernom zglobu (koji po definiciji ne prenosi rotacije između tijela). Eksterne sile djeluju na klizač i na nosač klizača tako da se moment $\mathbf{L}_3(t)$ u nosaču klizača (uslijed aktuatora u rotacijskom zglobu) suprotstavlja momentu koji nastaje uslijed sile $\mathbf{F}_4^1$ na klizaču. Sustav posjeduje jedan dinamički stupanj slobode: rotaciju u rotacijskom zglobu koja je dinamički kontrolirana varijabilnim momentom $\mathbf{L}_3(t)$ tako da rezultantno gibanje u zglobu posjeduje konstantnu kutnu akceleraciju. Klizač se po nosaču giba prema propisanom gibanju.

Slučaj 2 Slučaj je sličan slučaju 1 uz razliku da je sila na klizaču $\mathbf{F}_4^2$. U ovom se slučaju analizira zatajenje pogona klizača (koji nije opterećen punim opterećenjem, već postoji mala sila koja djeluje u pravcu translacijske osi klizača), tako da na njemu nema propisanog gibanja s obzirom na nosač klizača. Moment aktuatora rotacijskog zgloba je konstantan. Ovako definiran sustav posjeduje dva dinamička stupnja slobode gibanja

Slučaj 3 U odnosu na prethodna dva slučaja, slučaj 3 posjeduje velike rotacijske domene i služi kao pokazni primjer da se formulacijom numeričke integracijske metode na Lievoj grupi može integrirati dinamika gibanja sustava bez pojave kinamtičkih singulariteta, koji bi se nužno pojavili u slučaju bilo koje lokalne (vektorske) parametrizacije trodimenzijskih velikih rotacija. Sustav je definiran na isti način kao i slučaj 1, ali su kutne brzine propisanih gibanja definirane s $\mathbf{\Omega}_1^{G,2}$ i $\mathbf{\Omega}_2^{G,2}$. Isto kao i slučaj 1, broj dinamičkih stupnjeva slobode je jedan.

Integracijski algoritmi opisani u radu su implementirani u *MATLAB*-u zajedno s definicijom sustava. Dobivena rješenja su obrađena i uspoređena (za slučajeve 1 i 2) s rješenjima

dobivenim analizom sustava u *ADAMS*-u. Pokazuje se da rješenja dobivena formuliranim integracijskim metodama na Lievoj grupi ne odstupaju od referentnih rješenja, a manje razlike, koje proistječu iz upotrebe različitih integracijskih koraka, su objašnjene u radu. Rješenje dobivenom *ADAMS*-om uzeto je kao referentno točno rješenje kojem, smanjivanjem integracijskog koraka, konvergira rješenje na Lievoj grupi prostora stanja sustava

Peto poglavlje rada je zaključak gdje su sumirani svi parcijalni zaključci rada. U zaključku je pokazano kako su za slučaj 3 gibanja izbjegnuti višestruki slučajevi kinematičkih singulariteta koji bi zahtjevali dodatnu re-parametarizaciju dinamičkog modela u slučaju da su za opis matematičkog modela korištene standardne metode modeliranja i integracije diferencijalnih jednadžbi na vektorskim prostorima, a ne dinamički model definiran na Lievoj grupi prostora stanja sustava.

Chapter 1

Introduction

In this chapter an introduction to the dynamics of mechanical system is presented. The main problem of dynamics is stated, main theorems given and the procedure of modelling of such systems is briefly discussed. After the introduction in dynamics of mechanical systems and their modelling, a mathematical background required for the sequel of this thesis is presented and explained. More complex topics are also explained in some more details.

1.1 Dynamics of mechanical systems

Dynamics is a branch of classical mechanics that studies the motion of objects. The goal is to explain the motion of macroscopic objects acted upon by external forces. The starting point of this study is to specify the location of an arbitrary point, i.e. to find a suitable way of defining this. This leads to the introduction of metrics in space: the reference frame which enables us to describe the motion mathematically.

In order to be able to introduce some kind of mathematical description in space, firstly, the space itself is to be defined and described. In classical mechanics the assumption of space is that it is a three-dimensional construct (3D) and Euclidean [4], meaning that it is a flat non-curved space. The Euclidean space is denoted by $\mathbb{R}^n$ or $\mathbb{E}^n$, where n denotes the dimension of the space. In the case of three-space, the space is denoted with $\mathbb{E}^3$. Here the three-space is defined in short, but in later chapters spaces and metrics are defined and explained in more detail.

In the $\mathbb{E}^3$ space the position of a particle can be defined using three independent parameters: as the space dimension is three, three parameters define the position of the particle completely. Now, the concept of *distance* can be defined in different ways, also, the definition

of distance induces metric on the space. For example, if the Cartesian coordinate system, where three independent orthogonal unit vectors are defined, is used, the position of the particle is defined using three parameters giving the values of the distances along one unit vectors directions:

$$\vec{x} = \mathbf{x} = \sum_{i=1}^{3} x_i \vec{e}_i,$$

where $\vec{e}_i,\ i = \{1, 2, 3\}$ are the unit vectors defined earlier. If the particle is moving in time, then its position vector is a function of time t

$$\vec{x} = \vec{x}\,(t) = \sum_{i=1}^{3} x_i\,(t)\,\vec{e}_i.$$

This vector function, according to the notation above, can be written as three scalar functions of its components:

$$x_i = x_i\,(t)\,, \quad i = \{1, 2, 3\},$$

and this functions give the trajectory of the particle as functions of one parameter: time t. Obtaining the trajectory of particles and bodies is one of the main concerns of dynamics.

Returning to the induction of metrics in space, it is seen that the Cartesian coordinate system is introduced so that the definition of distance is:

$$\overline{xy} = d(x, y) = \sqrt{\sum_{i=1}^{3} (x_i - y_i)^2},$$

resulting in orthogonal unit vectors defining directions in $\mathbb{E}^3$ space.

Except position, which was the main topic till now, in the description of motion two additional properties play a crucial role: velocity and acceleration. Velocity is the first derivative of the position vector of a particle and it's also a vector quantity:

$$\mathbf{v}\,(t) = \frac{\mathrm{d}\mathbf{x}\,(\mathrm{t})}{\mathrm{dt}}.$$

Acceleration is the second derivative of the position vector or the first derivative of the velocity vector:

$$\mathbf{a}\,(t) = \frac{\mathrm{d}^2\mathbf{x}\,(\mathrm{t})}{\mathrm{dt}^2} = \frac{\mathrm{d}\mathbf{v}\,(\mathrm{t})}{\mathrm{dt}},$$

and it's a vector quantity.

Other properties of motion include momentum, angular momentum, kinetic and potential energy. There are more quantities that describe the motion, but the mentioned are among the most important ones.

In the following section degrees of freedom and the concept of generalized coordinates is explained.

1.1.1 System degrees of freedom and generalized coordinates

As it was described in the previous section, the motion of a particle or body is mathematically described using a certain number of parameters. The number of independent parameters that are required to completely define the position (configuration) of a dynamical system is called the number of *degrees of freedom* (DOF). For a dynamical system this independent parameters are time functions. The number of degrees of freedom is a property of the dynamical system.

For a free particle in three-space the number of parameters required for completely describing its position is three, $n = 3$, meaning that a particle in three-space has three DOFs. The number of DOFs is denoted with n. That implies that for a system with N free (unconstrained) particles the number of DOFs is equal to

$$n = 3N.$$

A free body in three space has $n = 6$ DOFs: three parameters defining the position of the body and three parameters that define the orientation of the body. Consequently, for N free bodies (unconstrained) in three-space the number of DOFs is equal to

$$n = 6N.$$

Going one step further, it can be stated that the number of DOFs defines the dimension of the space in which the motion "takes place". This means that the n-dimensional space contains all the information that is required for describing the motion and configuration of a dynamical system. All the calculations required for determining the motion are performed in that space. This n-dimensional space is called the *configuration space* of the system. Of course, if one is interested to describe the motion in more details more parameters can be used, but those new parameters are no longer independent.

The n-dimensional configuration space has for the coordinate basis the DOFs of the dynamical system, i.e. positions. From classical mechanics it is known that the velocity of a particle or point of a body is tangent to its trajectory. According to this, it can be concluded that velocities are vectors tangent to the configuration space at any point of that space. For specified initial conditions of the system only one trajectory, which is the solution of the

motion of the system, exists in the configuration space. At any point of that trajectory the tangent vector is the velocity at that point.

The position and velocity completely define the kinematics of the system, meaning that for the complete description of the system $2n$ parameters are required. Although acceleration is also present as a property of motion, it is usually known from the solution of algebraic equations, which originate from the 2^{nd} Newton Law where forces and accelerations are brought in direct correspondence. It is concluded that velocities and positions are the integration solution of the forward dynamics problem.

In dynamics, differential equations that need to be solved are of second order and they can be reformulated as a system of first order differential equations. The reformulated system of equations is the *state-space* formulation, which is solved for positions and velocities. Often the complete solution is not required, but just the relationship between velocities and positions is needed (for example energies in the system depend on positions and velocities). The plot of the relationship between the position and velocity (the $v\,(x)$ function) is called the *phase portrait* of the dynamical system and the $v - x$ plane is called the *velocity phase plane*.

The velocity phase space is formed out of displacements and velocities (related for each independent coordinate graphically), so that for a system of n DOFs its phase space is of dimension $2n$. Two-dimensional velocity phase portraits can be plotted for DOFs that can be separated out without involving other coordinates or for very simple single DOF systems.

Generalized coordinates

Each DOF of a dynamical system is mathematically described using one coordinate (number). *Generalized coordinates* are the minimal set of independent coordinates, whose number is equal to the number of DOFs and that completely describe the configuration of the system. The choice of coordinates used depends on geometrical properties of the system, shape, system topology, region of motion and other properties.

Earlier, the term *configuration space* was introduced and now its definition is extended. It was mentioned that the choice of generalized coordinates depends also on the region of motion of the system: the coordinates chosen must be able to describe all or most of the configurations that the system can achieve during its motion. The dimension of the configuration space is equal to the number of all the coordinates of the unconstrained system, i.e. to the sum of all bodies DOFs.

When constraints are imposed, the configuration space, whose dimension is equal to

the number of the system DOFs, is called *configuration manifold* (or *embedded manifold*, manifolds are explained later in the section 1.2) denoted with $\mathbb{Q}$. The generalized coordinates q_i, $i = \{1, \ldots, n\}$, lie on $\mathbb{Q}$ and

$$\dim \mathbb{Q} = n.$$

On figure 1.1 a manifold is graphically presented as a surface. Curves on that surface represent lines of constant q_i, i.e. this curves form the coordinate grid on the manifold. Also on the figure vectors tangent to the coordinate lines are depicted. Tangent vectors are defined and explained in section 1.2.4.

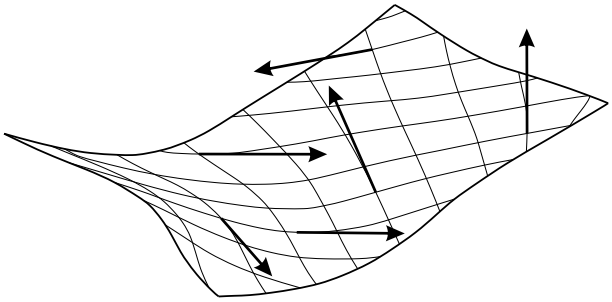


Figure 1.1: Manifold $\mathbb{Q}$ with its coordinate grid and tangent vectors [4].

The configuration manifold for a system with N unconstrained particles is the space of dimension $3N$ and it is the Euclidean space $\mathbb{E}^{3N}$, whose components are the $\mathbf{x}_i$, $i = \{1, \ldots, N\}$ coordinates of the particles. In the case of unconstrained bodies or particles, the configuration space and the configuration manifold are coincident.

1.1.2 Constraints

In mechanical systems constraints are common as bodies are interconnected and motions restricted in different ways (kinematic constraints), also different conserved quantities of a mechanical system form constraints in the mathematical sense. The introduction of kinematic constraints reduces the dimension of the configuration manifold as the constraints reduce the number of DOFs of the system.

Constraints induce a sub-manifold (*embedded manifold*) on the unconstrained configuration space. They constraint the system by defining a configuration hypersurface in $\mathbb{E}^{3N}$ (for a system of particles) and this hypersurface is the configuration manifold $\mathbb{Q}$. Also, it is seen that the system constraints curve the original configuration space which is a linear vector

space. On this new embedded manifold the motion takes place and this is the configuration manifold defined in the previous chapter.

A system of particles with K constraints possesses

$$n = 3N - K,$$

while a system of N rigid bodies with K constraints has

$$n = 6N - K$$

degrees of freedom. On the n-dimensional manifold $\mathbb{Q}$ the following coordinates are introduced:

$$\begin{aligned} q_i &= q_i\left(\mathbf{x}_1, ..., \mathbf{x}_N, t\right), \quad i = \{1, \ldots, 3N\}, \\ \mathbf{x}_k &= \mathbf{x}_k\left(q_1, ..., q_{3N}, t\right), \quad k = \{1, \ldots, N\}, \end{aligned}$$

which also define an invertible coordinate transformation valid for a region of $\mathbb{Q}$. The invertibility of the coordinate transformation is mathematically expressed via the *Jacobian matrix*, which has to be non-singular, i.e. its determinant J has to be

$$J = \det\left(\frac{\partial q_i}{\partial \mathbf{x}_i}\right) \neq 0,$$

where J is called the *Jacobian*.

The choice of generalized coordinates is related to the constraints of the system so that only the first n coordinates q_i are independent functions describing the system motion, while the last K of them reduce to trivial statements due to imposed constraints:

$$\begin{aligned} &q_{n+r} \neq q_{n+r}\left(t\right), \\ &q_{n+r} = R\left(0, \ldots, 0\right), \quad r = \{1, \ldots, K\}. \end{aligned} \tag{1.1}$$

This said, it can now be stated that the $q_{n+r},\ r = \{1, \ldots, K\}$, generalized coordinates define the configuration hypersurface (i.e. they constrain the system to $\mathbb{Q}$ on which the motion takes place and which is the so-called embedded manifold presented in section 3.1) while $q_i,\ i = \{1, \ldots, n\}$, represent the coordinates which define the actual position (configuration) of the system on the hypersurface (i.e. they lie on $\mathbb{Q}$). When generalized coordinates are referred to it is actually referred only to the q_i coordinates. In this way, the minimal form is defined: the number of coordinates is equal to the number of dynamic degrees of freedom and the governing (dynamics) equations are the equations of motion.

For a system of particles the complete point on $\mathbb{E}^{3N}$ is defined with both q_i and q_{n+r}, while the system position on $\mathbb{Q}$ is defined by coordinates q_i only. Instead of finding the complete solution $\mathbf{x}_k = \mathbf{x}_k(t)$ the problem is now reduced to finding the solution $q_i = q_i(t)$, which characterises the minimal form formulation.

The constraints are mathematically introduced as a set of algebraic or differential equations, depending on the type of constraint. Constraints can be holonomic, given in the implicit form by

$$f_r(\mathbf{x}_1, ..., \mathbf{x}_N, t) = 0, \quad r = \{1, \ldots, K < 3N\}, \tag{1.2}$$

which are integrable constraints. Nonholonomic constraints are non-integrable equations in the implicit form

$$f_r(\mathbf{x}_1, ..., \mathbf{x}_N, \dot{\mathbf{x}}_1, \ldots, \dot{\mathbf{x}}_N, t) = 0, \quad r = \{1, ..., K < 3N\}.$$

Furthermore, constraints can be divided by their time-dependence: constraints that are time dependent are called rheonomic while time-independent constraints are scleronomic.

When solving the system governing (dynamics) equations there is the need to simultaneously solve the constraint equations which have to be satisfied. This equations can be solved with the system dynamics equation by inclusion of the constraint forces. As the constraint forces are unknown, *Lagrange multipliers* are introduced together with the fact that, for ideal constraints (no friction), the constraint forces are perpendicular to the constraint surface. Using the statements above the i-th constraint force can be written as

$$Q_i^C = \lambda_i \nabla f_i(\mathbf{x}, t), \tag{1.3}$$

where $\nabla f_i(\mathbf{x}, t)$ (*nabla* operator, *gradient*) gives the vector perpendicular to the $f_i(\mathbf{x}, t)$ constraint surface. A Lagrange multiplier is denoted with λ_i and it describes the i-th constraint force magnitude. When the system governing equations are solved, the equations for the Lagrange multipliers are also solved.

In the case of ideal constraints, the constraint forces do no work, except in the case when the constraint surface moves in time. In the next section the system governing equations are discussed in more details and the system of equations, that needs to be solved in order to solve the forward dynamics problem, is established.

1.1.3 Modelling of dynamical systems

The first step in the modelling of dynamical mechanical systems is to set-up the mechanical model of the system, where bodies, particles and their relationships are established in

an as simple as possible way that still includes all the informations that are relevant and affect the results. During the mechanical modelling, bodies are simplified and standard constraints between them are set-up: the real-world system is discretized and all the important components are identified; simplifications and assumptions are made.

From the mechanical model the mathematical model is formulated, i.e. the mechanical model is the input for the establishment of the mathematical model. In classical mechanics Newton laws enable the formulation of a mathematical model that describes the mechanical model and its behaviour. By solving the basic set of equations the system motion and constraint forces are obtained. From [4] the two principle in mechanics are:

1. The inertial reference frame introduces time t as a linear parameter: time is linear with respect to length along the trajectory and is identical for all particles and bodies in the reference frame. This is equivalent to the 1^{st} Newton Law.

2. The conservation of momentum and the existence of mass enables the definition of the 3^{rd} Newton Law. The existence of mass enables the definition of force in the sense of the 2^{nd} Newton Law.

Here, the general form of the equations of motion is not derived and they are given as granted. The motion of a mechanical system is described using the *Newton-Euler equations*, presented here for one rigid body, in matrix form:

$$m\mathbf{a} = \mathbf{F}, \tag{1.4}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}\mathbf{J}\boldsymbol{\omega} = \mathbf{L}. \tag{1.5}$$

In the preceding equations the symbols involved are:

$\mathbf{a}$ Vector of acceleration at the body centre of gravity (CoG);

$\mathbf{F}$ Resulting force vector at the CoG;

$\boldsymbol{\omega}$ Angular velocity of the body (expressed in the body-fixed reference frame);

$\dot{\boldsymbol{\omega}}$ Time derivative of the angular velocity;

$\tilde{\boldsymbol{\omega}}$ Skew-symmetric matrix constructed from the vector of angular velocity;

$\mathbf{L}$ Resulting moment on the body (expressed in the body-fixed reference frame);

m Mass of the body;

J Inertia matrix of the body.

Equations (1.4) and (1.5) include all coordinates that give the position and orientation of the body, i.e. the time evolution of the position and orientation is obtained by solving this equations. This is an ODE system, which is rather straightforward to solve, but in the case of multibody system dynamics (MBS) constraints of the form (1.1) have to be included and then a different, extended, system of governing equations is obtained.

In the thesis the governing equations are formulated in the *descriptor* form, meaning that the number of coordinates (differential equations) is larger than the number of DOFs [9]. Usually, the number of coordinates in the equations is equal to the number of all the coordinates of all bodies in the system ($6N$ for a spatial system with N bodies).

The inertia matrix $\mathbf{M}$ is defined as

$$\mathbf{M}_i = \begin{bmatrix} m_i\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_i \end{bmatrix},$$

with the unitary matrix denoted with $\mathbf{I}$. Introducing the vector $\mathbf{v}_i = \begin{bmatrix} \dot{\mathbf{x}}_i & \boldsymbol{\omega}_i \end{bmatrix}^T$ equations (1.4) and (1.5) are written as a single matrix equation

$$\mathbf{M}_i\dot{\mathbf{v}}_i + \mathbf{Q}_i\left(\mathbf{x}_i, \mathbf{v}_i, t\right) + \mathbf{Q}_i^C\left(\mathbf{x}_i\right) = \mathbf{0}, \tag{1.6}$$

where $\mathbf{Q}_i\left(\mathbf{x}_i, \mathbf{v}_i, t\right)$ represents the external and non-linear velocity forces and $\mathbf{Q}_i^C\left(\mathbf{x}_i\right)$ are the constraint forces of the i-th body. It is important to note that the vector $\mathbf{x}_i$ is here composed of translation and rotation coordinates. For obtaining the complete mathematical model it is necessary to include information about constraint forces (equation (1.3)), system constraint equations (1.1) and sum the equations for all bodies in the system.

Defining the gradient matrix of the constraint functions as

$$\boldsymbol{\Phi}_x\left(\mathbf{x}\right) = \nabla\boldsymbol{\Phi}\left(\mathbf{x}, t\right) = \frac{\partial\boldsymbol{\Phi}}{\partial\mathbf{x}},$$

and, after introducing the Lagrange multipliers $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \dots & \lambda_K \end{bmatrix}^T$, the set of governing equations of the MBS is obtained in the form [9]

$$\begin{aligned} \mathbf{M}\dot{\mathbf{v}} + \mathbf{Q}\left(\mathbf{x}, \mathbf{v}, t\right) + \boldsymbol{\Phi}_x^T\left(\mathbf{x}\right)\boldsymbol{\lambda} &= \mathbf{0}, \\ \boldsymbol{\Phi}\left(\mathbf{x}\right) &= \mathbf{0}. \end{aligned} \tag{1.7}$$

The matrix $\boldsymbol{\Phi}\left(\mathbf{x}\right)$ is the collection of all constraint functions: $\boldsymbol{\Phi}\left(\mathbf{x}\right) = \begin{bmatrix} f_1\left(\mathbf{x}\right) & \dots & f_K\left(\mathbf{x}\right) \end{bmatrix}^T$.

Solving the set of equations (1.7) the system motion and constraint forces are obtained, but it is seen that this is not a set of ODEs. This set of equations constitutes a set of differential algebraic equations (DAEs) of dimension $6N+K$. There are 6 unknown functions per body and further K unknown Lagrange multipliers.

The obtained governing equations form the DAE system of index 3 because the constraint equations are at the displacement level. If the constraint equations (1.2) are formulated at the acceleration level, then the system of DAE of index 1 is obtained. The constraint equations at the acceleration level are, for holonomic systems, simply obtained by differentiating the set of constraint equations twice with respect to time:

$$\ddot{\mathbf{\Phi}}(\mathbf{x}) = \mathbf{0}.$$

The DAEs of index 1 are suitable for solving with the method presented in the thesis. The constraint equations at the acceleration level can be rewritten in the form

$$\mathbf{\Phi}_x \ddot{\mathbf{x}} = \boldsymbol{\xi},$$

which with the first equation of (1.7) form the DAE index 1 formulation of the governing equations

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_x^T \\ \mathbf{\Phi}_x & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \boldsymbol{\xi} \end{bmatrix}. \tag{1.8}$$

More details about the formulation of MBS governing equations can be found in the appendix of [10].

In the sequel of the thesis some more mathematical background is presented, especially in the part regarding manifolds and the group of rotations. After that, the integration procedure used for solving the set of equations (1.8) is described and the method demonstrated on a case study problem.

1.2 Manifolds

In this section manifolds are introduced and explained. Firstly manifolds are explained and then vector fields, vector spaces, curves and vector fields commutation are shortly described. The mathematics and terminology described here enables a better understanding of the following sections.

1.2.1 Manifold definition and properties

An n-dimensional manifold $\mathbb{M}$ is a construct that can be modelled in the real Euclidean vector space $\mathbb{R}^p$ whose dimension is $p \geq n$. It can be interpreted that $\mathbb{M}$ is a manifold if every point $m \in \mathbb{M}$ has an open neighbourhood which has a continuous *one-to-one (1-1)* map *"onto"* an open set of $\mathbb{R}^p$.

From the statements above it is concluded that a manifold is a construct that *locally* resembles (is locally "like") the Euclidean flat space $\mathbb{R}^n$, meaning that the local topology of $\mathbb{M}$ is the same as that of $\mathbb{R}^n$. It turns out that a manifold can be represented as a collection of this local charts defined in the $\mathbb{R}^n$ space at every point $m \in \mathbb{M}$.

The global topology of a manifold can be complex and different charts have to be used for different parts of the manifold.

More about maps

A map f is a rule that associates the element $x \in \mathbb{M}$ with an element $y \in \mathbb{N}$, $f : \mathbb{M} \to \mathbb{N}$. The spaces $\mathbb{M}$ and $\mathbb{N}$ don't have to be distinct as the mapping can map within the same space (e.g. a function that associates one point from $\mathbb{R}$ with another point in $\mathbb{R}$).

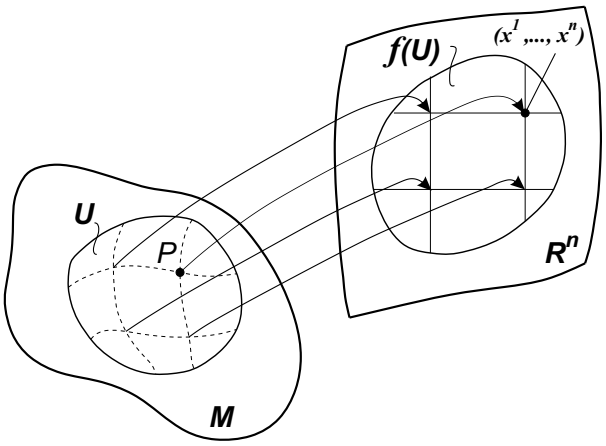


Figure 1.2: Mapping of an open neighbourhood of $\mathbb{M}$ *"onto"* $\mathbb{R}^n$ [8].

A *one-to-one* (1-1) map is a map that maps a point from one group (space) to one point in another group and there exists a unique inverse image of that mapping. If $f(x)$ is a 1-1 map that maps a point $x \in \mathbb{M}$ to a unique point $y \in \mathbb{N}$, then there exists the inverse f^{-1} which is again a 1-1 map.

If the map f is defined for all elements in $\mathbb{M}$ and the mapping has an inverse image (not necessarily unique) then f maps $\mathbb{M}$ *"onto"* $\mathbb{N}$.

In the context of manifolds a map f is a rule that associates to a point $m \in \mathbb{M}$, $\dim \mathbb{M} = n$, an n-tuple of real numbers $(x_1(m), \ldots, x_n(m))$ from $\mathbb{R}^n$:

$$f : \mathbb{M} \to \mathbb{R}^n,$$

and the numbers $x_1(m), \ldots, x_n(m)$ are called the coordinates of m under the map f. This is graphically shown on figure 1.2.

Charts

A *chart* is a pair consisting of the neighbourhood of a point $m \in \mathbb{M}$ and its map f. Another more formal definition states that a *local chart* is a map of a connected neighbourhood $U \subset \mathbb{M}$ (U subset of $\mathbb{M}$) which maps one point $u \in U$ to just one point of $\mathbb{R}^n$. In practice, the terms *chart* and *coordinate system* are equivalent: *chart* is just the mathematical jargon for saying *coordinate system*.

From the definition of a chart it is concluded that it covers only a portion of the manifold. In order to cover the whole manifold (all of its points) more charts have to be defined, or to reformulate, generally there is no single Cartesian-like coordinate system that covers the whole manifold and thus more overlapping n-dimensional Cartesian coordinate systems (called local charts) have to be used. The collection of all local charts of a manifold is called the *atlas*.

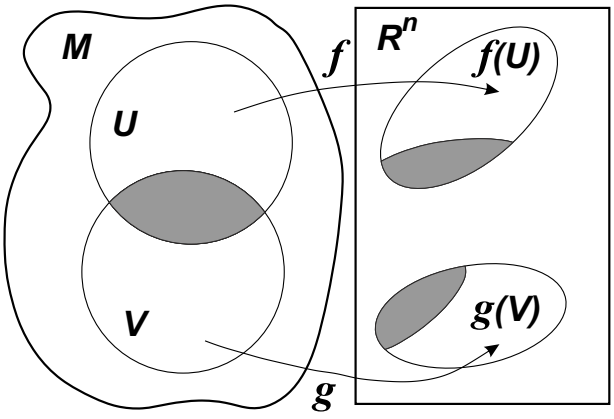


Figure 1.3: Overlapping of two regions U and V in $\mathbb{M}$ and their mapping *"onto"* $\mathbb{R}^n$ [8].

To be able to cover every single point of the manifold an open neighbourhood on the manifold must have overlaps with other open neighbourhoods in at least one chart. Figure

1.3 shows the overlapping of two neighbourhoods and their mapping *"onto"* $\mathbb{R}^n$. It is seen that the overlapping region is mapped twice.

Coordinate transformations can now be defined for overlapping regions of the manifold. From figure 1.4 two maps are defined:

$$f : U \in \mathbb{M} \to \mathbb{R}^n,$$
$$g : V \in \mathbb{M} \to \mathbb{R}^n,$$

where the coordinates of a point S from the overlap under f are $(x_1, \ldots, x_n)$ while the same point has coordinates $(y_1, \ldots, y_n)$ under the map g (figure 1.4).

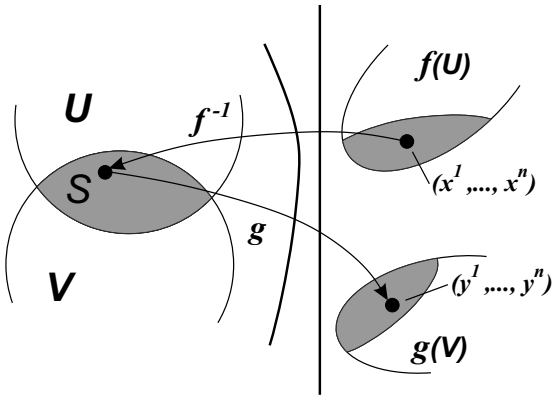


Figure 1.4: Mapping of the overlapping region and the coordinate transformation [8].

Using the two mappings defined above a composite map $\mathbb{R}^n \to \mathbb{R}^n$ can be constructed: by taking the inverse of the map f, a point in the overlap under that map is transferred back to the overlapping region on the manifold. This way the inverse map $f^{-1} : \mathbb{R}^n \to \mathbb{M}$ associates to the coordinates $(x_1, \ldots, x_n)$ a point $S \in \mathbb{M}$. The map g takes the point S to a point in $\mathbb{R}^n$ with coordinates $(y_1, \ldots, y_n)$. The above mentioned map is obtained:

$$g \circ f^{-1} : \mathbb{R}^n \to \mathbb{R}^n,$$

($g \circ f^{-1}$ is the composition of functions, it can be understood as $g\left(f^{-1}\right)$). The obtained functional relationship between the coordinates of the two mappings (charts) is called *coordinate transformation*:

$$y_1 = y_1\left(x_1, \ldots, x_n\right),$$
$$\vdots$$
$$y_n = y_1\left(x_1, \ldots, x_n\right).$$

If the transformation functions are C^k differentiable functions, then the manifold is called a C^k *differentiable manifold*. Consequently, if the transformation functions are differentiable so are their inverses. In mathematics the notation C^k means that a function is k-times differentiable. For a differentiable manifold it should be possible to construct a whole system of charts (an *atlas*) so that every $m \in \mathbb{M}$ is in at least one neighbourhood and every chart has to be C^k related to every other it overlaps with.

From [4] the definition of a manifold in terms of neighbourhoods and coordinate systems (i.e. local charts) is given:

> "A manifold is then a collection of points that is the union of a set of denumerable sets U_P in $\mathbb{R}^n$, each with its own local coordinate system Φ_P:
>
> $$\mathbb{M} = \cup_P U_P.$$"

1.2.2 Functions

A function on a manifold $\mathbb{M}$ is a rule that assigns a real number (called the value of the function) to each point of $\mathbb{M}$.

If m is a point on $\mathbb{M}$ which under a map g has coordinates $(x_1, \ldots, x_n)$ the function $f(m)$ can be written as $f(x_1, \ldots, x_n)$. The function is differentiable if it is differentiable in its arguments $x_1, \ldots, x_n$.

Functions are related to curves which are explained and detailed in the next section.

1.2.3 Curves

A curve is a differentiable mapping from an open set of $\mathbb{R}^1$ *"into"* $\mathbb{M}$. This is graphically presented on figure 1.5, and it is deduced that this is exactly the definition of a parametrically defined curve.

For a parametrically defined curve there is only one parameter λ, whose space is $\mathbb{R}^1$, and to every $\lambda \in \mathbb{R}^1$ the curve associates one point on the manifold $\mathbb{M}$, called the image point of λ. Mathematically, the curve on a manifold is given with a set of functions

$$\begin{aligned} x_1 &= x_1(\lambda), \\ &\vdots \\ x_n &= x_1(\lambda), \end{aligned}$$

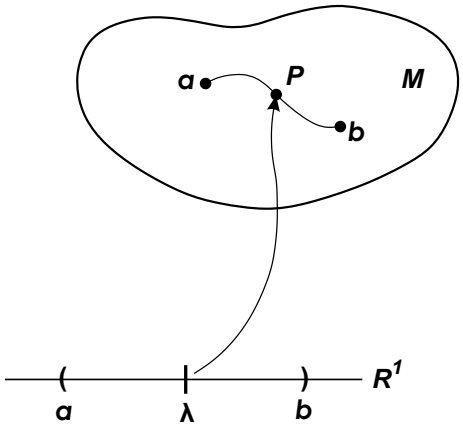


Figure 1.5: Curve as a map from $\mathbb{R}^1$ *"into"* $\mathbb{M}$ [8].

and if these functions are differentiable with respect to λ, then the mapping (i.e. the curve) is differentiable.

It is seen that for defining a curve on $\mathbb{M}$ there must exist a coordinate system defined in the neighbourhood of the curve, because the curve is defined as a set of coordinate functions of the parameter λ. Thus, the mathematical definition of a curve on a manifold is connected with the chart defined on the manifold.

1.2.4 Vector fields and vector spaces

Here the focus is laid upon vectors tangent to the manifold: vector fields and vector spaces are defined according to that. A vector tangent to the curve $x_i = x_i(\lambda)$, $i = \{1, \ldots, n\}$, on a manifold is defined in the following way. If f is a function, which has a value at each point of a parametric curve $x_i(\lambda)$, there exists a function g such that it gives values of f at the points defined by the parameter λ:

$$g(\lambda) = f(x_1(\lambda), \ldots, x_n(\lambda)).$$

Differentiating g yields:

$$\frac{\mathrm{dg}}{\mathrm{d}\lambda} = \sum_{i=1}^{n} \frac{\mathrm{dx_i}}{\mathrm{d}\lambda} \frac{\partial f}{\partial x_i},$$

and that is true for any function g so that the operator $\frac{\mathrm{d}}{\mathrm{d}\lambda}$ is defined as

$$\frac{\mathrm{d}}{\mathrm{d}\lambda} = \sum_{i=1}^{n} \frac{\mathrm{dx_i}}{\mathrm{d}\lambda} \frac{\partial}{\partial x_i}. \tag{1.9}$$

In equation (1.9) the terms $\frac{\mathrm{dx_i}}{\mathrm{d}\lambda}$ are components of a vector tangent to the curve $x_i(\lambda)$. Those terms are a set of coordinates whose coordinate basis (vector basis) is defined with $\frac{\partial}{\partial x_i}$ which represents the derivations along the coordinate lines.

According to this, the directional derivatives along curves at a point $m \in \mathbb{M}$ form a vector space whose basis are the directional derivatives $\frac{\partial}{\partial x_i}$. In this way the relationship between tangent vectors and derivatives along curves on manifolds is established. The space of all tangent vectors at m and the space of derivatives along all curves at m are in 1-1 correspondence. This implies that at $m \in \mathbb{M}$ there is only one tangent vector space containing tangent vectors of all curves through m.

The tangent vector space at the point m of the manifold is denoted with T_m. In general, the manifold is not a linear space, it's rather a curved hypersurface as it was already mentioned above. A property of a vector space is that vectors belonging to the same vector space can be added one to another: vectors belonging to T_m can be added one to another. Let's denote the tangent vector space at a point $p \in \mathbb{M}$ with T_p. As m and p are different points on the same manifold, vectors from T_m and those from T_p cannot be added one to another because they belong to different tangent vector spaces. By thinking of a manifold as a curved surface, that fact is easily visualized: tangent vector spaces at different points are planes that are not generally parallel one to another and, consequently, vectors from different spaces are in no linear relation. This is a very important fact that should be kept in mind when thinking of vectors in tangent spaces. This is graphically shown later in section 1.3.1 where the *tangent bundle* is defined.

Returning to the tangent vector definition, it has to be mentioned that a tangent vector doesn't have a unique curve to which it is tangent. One vector in T_m is tangent to an infinite number of curves:

- There are many curves through m tangent one to another and have the same tangent vector;
- The same path on the manifold may be reparametrized in such a way that the tangent at m is retained the same.

A *vector field* is a rule for defining a vector at each point of $\mathbb{M}$. Moreover, it is a rule that assigns a tangent vector to each point $m \in \mathbb{M}$ which lies on the tangent space T_m. The vector field selects one vector from each tangent vector space.

Above it was stated that every curve on $\mathbb{M}$ has a tangent vector at every point it passes through. If a point $m \in \mathbb{M}$ is known (given by the initial conditions) along with a vector

field Δ it is possible to find a curve starting at m and such that its tangent vector at any point of the curve always belongs to the vector field. The vector field has to be at least C^1 differentiable and the obtained curve is called the *integral curve*. In the section 1.3 it is presented that the governing equations of a mechanical system define a vector field and it follows that integral curves are the solution of the system dynamics. Also, some other properties and the way of looking on vector fields are presented.

1.2.5 Vector fields commutation

The *commutator* is an operator defined for two operators X and Y as

$$[X, Y] = XY - YX. \tag{1.10}$$

If X and Y are operators on a function f, then the commutator can be written as

$$[X, Y](f) = X(Y(f)) - Y(X(f)).$$

The commutator [,], when operated on vector fields, is called the *Lie bracket*. If $\vec{V}$ and $\vec{W}$ are two vector fields

$$\vec{V} = \frac{\mathrm{d}}{\mathrm{d}\lambda},$$
$$\vec{W} = \frac{\mathrm{d}}{\mathrm{d}\mu},$$

then the commutator of these vector fields is

$$\left[\vec{V}, \vec{W}\right] = \vec{V}\vec{W} - \vec{W}\vec{V}.$$

The vector fields are said to commute if the Lie bracket equals to zero, i.e. $\left[\vec{V}, \vec{W}\right] = 0$. If the commutator is $\left[\vec{V}, \vec{W}\right] \neq 0$ the two vector fields don't commute. In general the Lie bracket of two vector fields is again a vector field and from the discussion above it is seen that generally two vector fields don't need to commute. Mathematically, the fact that two arbitrary vector fields defined above don't commute can be stated the following way:

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\frac{\mathrm{d}}{\mathrm{d}\mu} \neq \frac{\mathrm{d}}{\mathrm{d}\mu}\frac{\mathrm{d}}{\mathrm{d}\lambda}.$$

This is graphically shown on figure 1.6: for an arbitrary displacement ϵ along the curves of the vector basis it is not the same in which order the displacements are performed. If, firstly, $\delta\lambda = \epsilon$ (displacement along $\frac{\mathrm{d}}{\mathrm{d}\lambda}$) and then $\delta\mu = \epsilon$ (displacement along $\frac{\mathrm{d}}{\mathrm{d}\mu}$) is performed

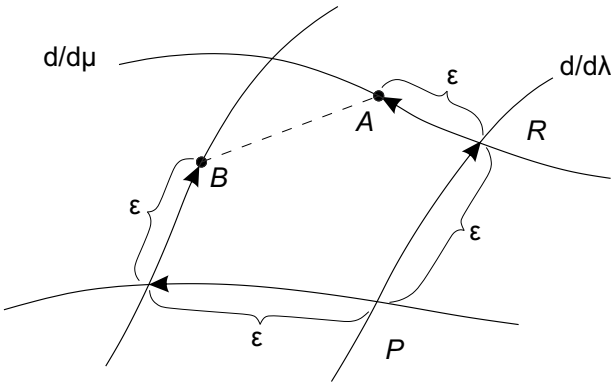


Figure 1.6: Graphical description of vector field commutation [8].

one ends up at A. If the order is changed, firstly, the displacement $\delta\mu$ is performed and then $\delta\lambda$, the resulting point is B. This shows that the two vector fields don't commute.

The vector fields generally don't commute because the derivative $\frac{\mathrm{d}}{\mathrm{d}\lambda}$ is not a derivative with fixed μ and $\frac{\mathrm{d}}{\mathrm{d}\mu}$ is not a derivative with fixed λ. This means that an integral curve of the vector field $\vec{V}$ is not necessarily a curve of constant μ, the same is valid for integral curves of $\vec{W}$ and λ. This can be understood from the fact that λ and μ are parameters rather than coordinates.

Coordinate basis $\frac{\partial}{\partial x_1}$ and $\frac{\partial}{\partial x_2}$ commute because the basis are formed in such a way that an integral curve x_1 is a curve of constant x_2 and vice versa. As x_1 is constant along x_2 (the inverse is also true) the two basis commute:

$$\frac{\partial}{\partial x_1}\frac{\partial}{\partial x_2} = \frac{\partial}{\partial x_2}\frac{\partial}{\partial x_1}.$$

It can be shown, and it is deducible from the discussion above, that two vector fields form a coordinate basis if they commute.

In the following section one step towards the application of the terms and constructs defined is made. The dynamics of multibody systems is presented in terms of manifolds and constructs of manifolds.

1.3 Manifolds in multibody system dynamics

In section 1.1.1 and later sections the idea of the configuration manifold was introduced and generalized coordinates were defined. What is concluded from that discussion is that the configuration manifold $\mathbb{Q}$ is a collection of points where each point represents one configuration (position) of the system. Later it was discussed that the tangents to curves on

manifolds, where each curve now represents a trajectory of the system with time t as the parameter, exist in tangent vector spaces (at a point $q \in \mathbb{Q}$ the tangent space is T_q). More details about how the configuration manifold (or embedded manifold) is obtained in the configuration space are given in section 3.1.

1.3.1 The tangent bundle

In dynamics positions and velocities play a very important role and it is seen that they constitute a manifold and vector space respectively. On the configuration manifold there exists an infinite number of trajectories of the system: every trajectory (curve on the manifold) represents one possible physical motion of the system and its tangent vectors represent velocities.

To each point q of the configuration manifold $\mathbb{Q}$ the tangent vector space T_q can be associated. The collection of all the points of the configuration manifold together with the tangent vector spaces at that points leads to the definition of a larger manifold called the *tangent bundle* denoted with $T\mathbb{Q}$. The dimension of this manifold is $2n$ where n is the number of the system DOFs (and, therefore, the dimension of the configuration manifold $\mathbb{Q}$, which is called the *base manifold* of the tangent bundle). The tangent bundle involves generalized coordinates q_i and generalized velocities $\dot{q}_i$ and it is also referred to as the *fibre bundle*.

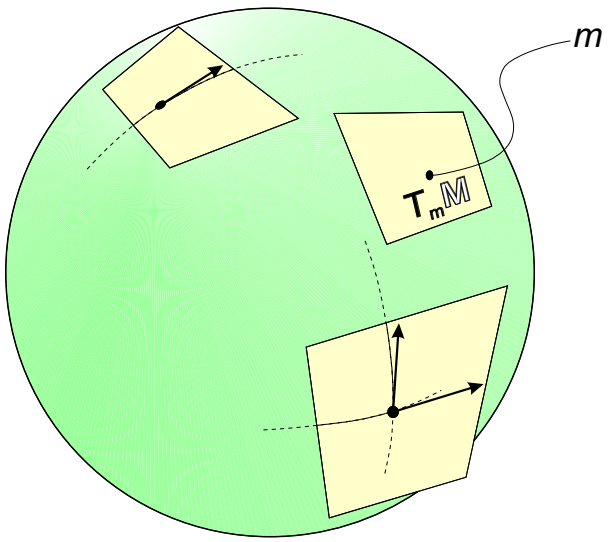


Figure 1.7: The $\mathbb{S}^2$ sphere as a manifold with tangent spaces [4].

The most simple example of a tangent bundle is the velocity phase space of dimension 2:

the motion of a particle along a curve is a 1-dimensional motion and by adjoining to each point of the trajectory a vector space (again a 1-dimensional object) a 2-dimensional flat space is obtained known under the name *velocity phase plane*. Each point on that tangent bundle associates exactly one velocity vector to one point of the trajectory.

Figure 1.7 shows the sphere $\mathbb{S}^2$ as a manifold with a few tangent spaces. The collection of all tangent spaces at every point on the sphere with the base sphere manifold constitutes the sphere tangent bundle $T\mathbb{S}^2$. The dimension of the sphere is 2 and the tangent space is a plane also of dimension 2, so the resulting tangent bundle is of dimension 4 ($n = 2$, $\dim T\mathbb{S}^2 = 2n = 4$). Here, it is seen that the discussion of section 1.2.4 holds and is graphically shown that two vectors from different tangent spaces cannot be added one to another.

Now, to formalize the discussion above, the construction of the tangent bundle is formulated:

> At each point $q \in \mathbb{Q}$ there exists a vector space called the tangent space $T_q\mathbb{Q}$ containing all possible velocities at q. As velocities $\dot{q}$ are tangent to the trajectories at q, so is $T_q\mathbb{Q}$ tangent to $\mathbb{Q}$ at that same point.
>
> The tangent bundle $T\mathbb{Q}$ is obtained when all tangent spaces $T_q\mathbb{Q}$ are adjoined to the base manifold $\mathbb{Q}$, $\forall q \in \mathbb{Q}$:
>
> $$T\mathbb{Q} = \cup T_q\mathbb{Q}, \quad \forall q \in \mathbb{Q}.$$
>
> This way $T\mathbb{Q}$ is composed of $\mathbb{Q}$ plus all the tangent spaces $T_q\mathbb{Q}$ of every point q on the base manifold.

Finally, without going into too many details, the tangent (fibre) bundle $T\mathbb{Q}$ is a space for which it is given:

- The base manifold $\mathbb{Q}$;
- A projection map $\Pi : T\mathbb{Q} \to \mathbb{Q}$;
- A typical fibre F,

where the typical fibre F is the vector space that gives, i.e. has, the same structure as the tangent space.

1.3.2 Properties of tangent bundles

Properties of the tangent bundle are mainly inherited from manifold and vector space properties. From the discussion of section 1.2.4, it follows that elements belonging to different tangent spaces (tangent spaces defined at different points of the base manifold) cannot be added directly. This leads to the conclusion that elements on a single fibre T_q can be added directly one to another, while elements on different fibres cannot.

Also, as the vector field associates to each point of the base manifold a tangent vector, it follows that the vector field is a mapping from the base manifold *"into"* the tangent bundle.

The tangent bundle is a special kind of manifold since it is decomposable into fibres (tangent spaces) and there exists a defined projection map which maps any point of the fibre to the point of the base manifold to which the fibre is adjoined to.

Finally, one important property of the tangent bundle is that it is, like the base manifold, a differentiable manifold.

1.3.3 Dynamics on tangent bundles

In section 1.2.4, it was mentioned that an integral curve, obtained from a known vector field and given initial conditions, defining a starting point on the configuration manifold, presents a solution of the system dynamics.

It can be shown that the equations of motion of the mechanical system are equivalent to a vector field Δ on the manifold $T\mathbb{Q}$; the EOMs give the system accelerations as functions on the manifold. When formulating the system EOMs using the Lagrange equations, it is also seen that the *Lagrangian* L defined as

$$L\left(q, \dot{q}\right) = T - V,$$

where T is the kinetic energy and V is the potential energy, is a function on $T\mathbb{Q}$ as both the potential and kinetic energies are functions of generalized coordinates and generalized velocities.

From this short discussion it is seen that the dynamics of a system exists on a manifold and all the functions are functions on that manifold. The usual approach in mechanics is to use a local chart and solve the problem in the neighbourhood of the point where the problem is defined. For problems with greater neighbourhoods this is not so simple as singularities in the charts used arise and some positions (configurations) of the system cannot be described.

This is a problem often encountered in the description of rigid body rotations, especially when rotations are greater than the domain of the used parametrization.

In the following chapter rotations and their mathematical description in terms of groups and manifolds is presented. Until now no formal mathematical mention of the rotations was made and the dynamics was mostly focused on particles and body translations. After the detailed presentation of rotations of the following chapter the integration method is presented and the solver algorithm given.

Chapter 2

Rotations and the $SO(3)$ group

A *rotation* is a kind of displacement defined for bodies (objects consisting of more particles) in which a point (not necessarily belonging to the body) is not displaced and that point is called the centre of rotation [6].

The terms *rotation* and *orientation* have distinct meanings: while rotation is a process, orientation is a state of the object. A rotation can be seen either as the change in orientation, or a relation between two orientations of the object.

Rotations can be composed and in general do not commute. It is important to mention that, although infinitesimal rotations are seen as vectors and thus their composition is pure vector addition, finite rotations are not vector quantities and their composition is not addition.

2.1 Mathematical description of rotations

The orientation of an object can be determined by providing coordinates of some vectors attached to the object, so that, for a p-dimensional object in n-dimensional space, the orientation matrix $\mathbf{O}$ is of dimension: $\dim(\mathbf{O}) = p \times n$ [6]. This matrix is formed of p vectors with n elements. As in three-dimensional (3D) space the object considered are bodies of the same dimension in the sequel it holds $p = n$.

The matrix $\mathbf{O}$ gives the orientation of the body and, by specifying some properties of the vectors that form the matrix, special properties and groups are defined.

For $p = n$ the matrix $\mathbf{O}$ is a square matrix and, if the n vectors in the matrix are mutually orthogonal, the matrix is an orthogonal matrix which satisfies the orthogonality condition

$$\mathbf{O}\mathbf{O}^T = \mathbf{I},$$

where $\mathbf{I}$ is the unitary matrix, $\dim(\mathbf{I}) = n \times n$. If the vectors in the matrix are additionally of unit magnitude all the matrices $\mathbf{O}$ constitute a topological space that form the base of the so-called *Stiefel manifold*.

Rotations are composed by simple matrix multiplications corresponding to scalar products of the vectors in the rotation matrices $\mathbf{O}$ and $\mathbf{O}_1$ that represent two orientations of the body:

$$\mathbf{R} = \mathbf{O}_1\mathbf{O}^T.$$

The matrix $\mathbf{R}$ determines the rotation of the body from the orientation $\mathbf{O}$ to the orientation $\mathbf{O}_1$ and it can be interpreted as the transformation from the zero orientation to the orientation given with $\mathbf{O}_1$. As the orientation matrix $\mathbf{O}$ is composed of vectors, the rotation matrix $\mathbf{R}$ can be seen as a transformation on vectors. The composition of consequent rotations is again simple matrix multiplication:

$$\begin{aligned}
\mathbf{R}_1 &= \mathbf{O}_1\mathbf{O}^T, \\
\mathbf{R}_2 &= \mathbf{O}_2\mathbf{O}_1^T, \\
\mathbf{R} &= \mathbf{R}_2\mathbf{R}_1 = \mathbf{O}_2\mathbf{O}^T, \\
\mathbf{O}_2 &= \mathbf{R}_2\mathbf{R}_1\mathbf{O},
\end{aligned}$$

where the orthogonality condition $\mathbf{O}_1\mathbf{O}_1^T = \mathbf{I}$ is used. The composition of rotations is associative:

$$(\mathbf{R}_3\mathbf{R}_2)\,\mathbf{R}_1 = \mathbf{R}_3\,(\mathbf{R}_2\mathbf{R}_1)\,.$$

The determinant of the matrix $\mathbf{R}$ is

$$\det(\mathbf{R}) = \det\left(\mathbf{O}_1\mathbf{O}^T\right) = \det(\mathbf{O}_1)\det(\mathbf{O})\,,$$

and from the properties of orthogonal matrices it follows that

$$\det(\mathbf{R}) = \pm 1.$$

It can be shown that, when the determinant of the rotation matrix is $\det(\mathbf{R}) = -1$, the object changes handedness, meaning that its mirrored image is obtained. In mechanics, where real bodies are regarded, this is an impossible transformation and consequently rotations are limited to be such that $\det(\mathbf{R}) = +1$. This way the orientations are determined by orthogonal matrices of the same determinant sign.

In the Euclidean three-space ($\mathbb{R}^3$) to describe the orientation/rotation of a body three vectors with three components are required. It follows that $\dim(\mathbf{O}) = 3 \times 3$ and, using the

orthogonality property of those matrices, it can be shown that, for proper rotations in $\mathbb{R}^3$, only three independent parameters are required to give the body orientation. This gives the opportunity to parametrize the rotation of a body with three parameters. In section 2.2.1 more information about parametrizations of rotations is presented.

2.1.1 Groups

Before explaining why rotations form a group, the mathematical definition and explanation of groups is presented in order to understand and be able to draw conclusions about the rotations group. The understanding of groups and manifolds is very important for the later study of Lie groups in section 2.3.2.

A group $\mathbb{G}$ is a collection of elements together with a binary operation $\cdot$ if the following axioms are satisfied [8]:

1. Associativity

$$x, y, z \in \mathbb{G},$$
$$x \cdot (y \cdot z) = (x \cdot y) \cdot z;$$

2. There must exist an identity element

$$x \cdot e = x;$$

3. Inverse

$$x \cdot x^{-1} = e,$$
$$x^{-1} \cdot x = e;$$

4. If in addition

$$x \cdot y = y \cdot x,$$

   the group is commutative (*Abelian*).

The result of the group binary operation $\cdot$ between two elements of the group is again an element of the group. This is called *closure*: $x, y \in \mathbb{G}$, then $x \cdot y \in \mathbb{G}$.

More group properties

Some other terms used for describing groups and their properties have to be introduced.

Two groups $\mathbb{M}_1$ and $\mathbb{M}_2$ with their binary operation $\cdot$ and $*$ are *homomorphic* if there is a map of $\mathbb{M}_1$ *"into"* $\mathbb{M}_2$ which respects the group operations [8]:

$$f(x \cdot y) = f(x) * f(y). \tag{2.1}$$

Homomorphism is a structure preserving map and it can be a *many-to-one* (maps many elements of one group to one element of another) and only *"into"* (*"into"* meaning that the inverse image is not required to exist as in an *"onto"* mapping). If the map is a 1-1 and *"onto"* then the two groups are *isomorphic*, meaning that the two groups are identical in their group properties. Isomorphic groups are structurally identical and there is a full correspondence between the elements and operations of the groups, also, the equation (2.1) has to be satisfied.

It is seen that homomorphism is a wider term than isomorphism. Going into more detail, if the isomorphism as a map is in addition C^∞ differentiable the groups are then *diffeomorphic*. In this context mappings follow from groups and map properties are inherited from the group properties. Here groups and manifolds are closely related.

2.2 Rotations as a group

A *subgroup* can be formed by grouping some elements of a group with the same binary operation, which themselves form a group (i.e. satisfy group axioms). The subset of the group (subgroup) always has the same properties as the larger group.

The set of rotations, with composition as the operation between them, constitutes a group isomorphic the the group of orthogonal $n \times n$ matrices and this group is denoted with $O(n)$. This group comprises all possible rotations of an object. As only real 3D bodies are considered the matrices are of dimension 3×3 (implying $n = 3$).

The matrices $\mathbf{R}$, with $\det(\mathbf{R}) = +1$, are called *special orthogonal matrices* and form a group of 3×3 matrices isomorphic to the *special orthogonal group* of matrices denoted with $SO(3)$ [6]. The characteristic of this group is that the vector scalar product (vector inner, dot, product) is a preserved quantity and the group is closed under compositions (matrix multiplication) and taking inverses. This means that the $SO(3)$ group is an invariant subgroup of $O(3)$ in which the inner product is a conserved quantity:

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} : \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = +1 \right\}. \tag{2.2}$$

From above, it follows that the $SO(3)$ group is a subgroup of $O(3)$ ($SO(3) \leq O(3)$, constrained from $O(3)$ with $\det(\mathbf{R}) = +1$). From equation (2.2) it is seen that the $O(3)$ group is defined by the orthogonality condition as the constraint on the $GL(3)$ group (general linear group of 3×3 invertible matrices). As the result of the group operation between two group elements is again in an element of the group, it is concluded that $O(3)$, and therefore $SO(3)$, are topological spaces.

Going into more details it can be said that the $SO(3)$ group is formed from the $GL(3)$ group with the constrains of equation (2.2) which define a hypersurface in $GL(3)$. This, according to the discussion from sections 1.1.2 and 1.2.1, implies that $SO(3)$ is also a manifold.

The matrices from the special orthogonal group describe proper rotations (rotations with positive determinants, no mirroring or inversion) and there is a 1-1 correspondence between special orthogonal matrices and proper rotations.

From the definition of groups (section 2.1.1), it follows that every group must have an identity element (e), which in the case of rotations is the identity rotation given with the unitary matrix $\mathbf{I}$. The identity rotation doesn't change the orientation for any arbitrary $\mathbf{R}$:

$$\mathbf{IR} = \mathbf{RI} = \mathbf{R}.$$

Returning to the topology of the rotation group, it should be added that the group of all rotations $O(3)$, that includes proper and improper rotations, consists of two disjoint open subsets (two subsets defined with different determinants: $\det(\mathbf{O}) = \pm 1$, $\mathbf{O} \in O(3)$). In contrary, the group of proper rotations $SO(3)$ is a connected group and there is always a path on the group that connects one orientation to the subsequent one.

2.2.1 Rotation parametrizations

Now a little digression towards the parametric description of rotations is made. Also the problems that arise with parametrizations are presented.

A parametrization is seen as a mapping from the $SO(3)$ group *"onto"* the Euclidean $\mathbb{R}^3$ space. This can be thought of as a coordinate system on the $SO(3)$ manifold (which, according to the discussion of section 1.2, is not able to cover the whole manifold). Because there is no single parametrization that is 1-1, continuous and with a continuous inverse on whole $SO(3)$ different parametrizations are used. Some types of parametrizations are:

- Rodrigues parameters;

- Euler angles;
- Cayley-Klein parameters;
- Quaternions;
- Rotation vector.

Details regarding the types of parametrization are not presented as they are not important for the scope of this thesis, but for the calculation of initial rotation matrices in chapter 4 a parametrization has to be used. For that purpose Euler angles (313 rotation sequence) θ_1, θ_2 and θ_3 are used as parameters and the rotation matrix is calculated using

$$\mathbf{R} = \begin{bmatrix} c_1c_3 - c_2s_1s_3 & -c_1s_3 - c_2c_3s_1 & s_1s_2 \\ c_3s_1 + c_1c_2s_3 & c_1c_2c_3 - s_1s_3 & -c_1s_2 \\ s_2s_3 & c_3s_2 & c_2 \end{bmatrix}, \tag{2.3}$$

where

$$\begin{aligned} s_1 &= \sin\theta_1, & c_1 &= \cos\theta_1, \\ s_2 &= \sin\theta_2, & c_2 &= \cos\theta_2, \\ s_3 &= \sin\theta_3, & c_3 &= \cos\theta_3. \end{aligned}$$

It is important to mention that parametrizations differ in the type of information they use to describe rotations. Different parametrizations have different domains in which the parametrization is 1-1, continuous and with a continuous inverse.

According to the above mentioned, in every parametrization singularities occur - the map is not any more 1-1 and there is no inverse. In order to be able to describe rotations that are greater than the domain of the parametrization used, more than one parametrization has to be used. The final result is then affected by errors (the rotation matrix is no longer orthogonal) and the correction of errors is a computationally costly process as it consists of finding the "nearest" orthogonal matrix (orthonormalization of a matrix). Also, the change from one parametrization to another is not convenient and complicates the calculations and solution procedure.

Here lies the main advantage of a geometric integration procedure directly on the $SO(3)$ group: for large rotations the need for re-parametrization is avoided, meaning that singularities are avoided (as they arise only on local charts, not on the group itself - the group is continuous).

2.2.2 Euler's and Chasles's theorems

Before presenting the $SO(3)$ group, the *Euler's* and *Chasles's theorems* are presented as they provide means for better understanding of the methods and procedures presented in the thesis. They give a different point of view on rigid body spatial rotations and general body displacements.

As three-dimensional rotations can be parametrized with three parameters, one can see a rotational transformation as a vector quantity. The *Euler's theorem* states [1]:

> "The most general displacement of a rigid body with one point fixed can be described as a single rotation about some axis through that fixed point, called the *axis of rotation* or *principal line*."

It's important to keep in mind that two consequent rotations represented by Euler rotation vectors cannot be composed by simple vector addition. In the sequel it's shown that this rotation vector exists in the $so(3)$ Lie algebra (section 2.3.3) of the $SO(3)$ Lie group (section 2.3.2) and this is an important property used in the integration method.

For completeness, the *Chasles's theorem*, which extends the Euler's theorem to the general rigid body spatial displacement, is also given [1]:

> "The most general displacement of a rigid body is equivalent to a translation of some point in the body, plus a rotation about an axis through that point."

This theorem describes a general body displacement as a screw motion.

Another approach to the integration of MBS motion is by using the Chasles's theorem, which leads to modelling and integration of systems on the $SE(3)$ group. This is beyond the scope of this thesis and it's just mentioned as another direction of research and methods formulation.

2.3 The $SO(3)$ group

In this section properties and mathematical constructs of the $SO(3)$ group, which provide the framework for direct integration on the group, are presented. Firstly the $SO(3)$ group is described as a differentiable manifold and later *Lie groups* and the *Lie algebra* are introduced along with *exponential mapping*.

2.3.1 $SO(3)$ as a differential manifold

The $SO(3)$ group can be seen as a differentiable manifold covered by charts (parametrizations) which are partially overlapping. This is seen from the $SO(3)$ definition of (2.2) where, firstly, $SO(3)$ is defined as a manifold embedded in $GL(3)$ with the orthogonality constraint. As it was already mentioned and can be concluded from section 2.2.1, the chart φ maps an open sets of the $SO(3)$ manifold to an open set of $\mathbb{R}^3$: $\varphi : SO(3) \to \mathbb{R}^3$.

The differentiability of $SO(3)$ enables the definition of tangents to the manifold and, also, the vector of angular velocity can be seen as a tangent vector on the $SO(3)$ manifold, but a few additional comments are required to completely understand the angular velocity on $SO(3)$.

Because infinitesimal angles of rotation are regarded as vector quantities, the angular velocity (rate of change of the orientation) is also a vector. As the angular velocity is obtained from the derivation of the rotation matrix with respect to time and the rotation matrices belong to $SO(3)$, then it is concluded that angular velocity is also a vector on the $SO(3)$ manifold, i.e. in its tangent vector spaces. At a point $m \in SO(3)$ the angular velocity is a vector on $T_mSO(3)$, but its components don't represent mathematically the angular velocity (as it's not a skew-symmetric matrix). The mathematical representation of the angular velocity is when that vector on the tangent space $T_mSO(3)$ is mathematically transferred back to the unity of the group (where the *Lie algebra* is defined, more in section 2.3.3)

Additionally, the $SO(3)$ manifold can be regarded as a *Riemannian manifold* for which a metric is specified, meaning that a smooth field of the symmetric and positive definite metric tensor g_{ij} is specified. At a point $m \in SO(3)$ the metric tensor field sends vectors $\vec{v}$ and $\vec{w}$ from the tangent space $T_mSO(3)$ to real numbers. This is a generalization of the vector inner (scalar) product. From the definition of a Riemannian manifold it is seen that $SO(3)$ satisfies the requirement: a Riemannian manifold is a real differentiable manifold in which each tangent vector space is equipped with an inner product g_{ij} called a *Riemannian metric* which varies smoothly from point to point.

2.3.2 $SO(3)$ as a Lie group

Before going into details what makes the $SO(3)$ group a *Lie group*, the deinition of a *Lie group* is given.

A *Lie group* $\mathbb{G}$ is a group which is also a differentiable manifold of finite dimension and

whose group operator and inversion are smooth maps. The group operator f is a smooth mapping from the manifold *"onto"* itself: $f : \mathbb{G} \times \mathbb{G} \to \mathbb{G}$. The definition of a Lie group from [3] is:

> "A *Lie group* is a differential manifold $\mathbb{G}$ equipped with a product $\cdot : \mathbb{G} \times \mathbb{G} \to \mathbb{G}$ satisfying
>
> $p \cdot (q \cdot r) = (p \cdot q) \cdot r \quad \forall\, p, q, r \in \mathbb{G}$ (associativity),
>
> $\exists\, \mathbf{I} \in \mathbb{G}$ such that $\mathbf{I} \cdot p = p \cdot \mathbf{I} \quad \forall\, p \in \mathbb{G}$ (identity element),
>
> $\forall\, p \in \mathbb{G}\ \exists\, p^{-1} \in \mathbb{G}$ such that $p^{-1} \cdot p = \mathbf{I}$ (inverse),
>
> the maps $(p, r) \mapsto p \cdot r$ and $p \mapsto p^{-1}$ are smooth functions (smoothness)."

Additionally, from [8], the definition of a Lie group states that it is a C^∞ manifold $\mathbb{G}$ of finite dimension

> "... which has the following C^∞ maps (diffeomorphisms): any element g of $\mathbb{G}$ maps $h \mapsto gh$ (*left translation by* g) or $h \mapsto hg$ (*right translation by* g)."

Lie groups are named after the Norwegian mathematician *Marius Sophus Lie*, who largely created the theory of continuous symmetry and applied it to geometry and differential equations.

From the definition of a Lie group it is seen that a Lie Group is, firstly, a manifold and then, secondly, points of the manifold are group elements. $SO(3)$ is such a construct:

1. Firstly, it is a manifold: an entity of complex structure defined with (2.2) that can locally be represented in a coordinate system (in the case of rotations with a parametrization).

2. Secondly, the points (elements) of the manifold are rotation matrices that themselves form a group with matrix multiplication (composition of rotations) as the group operation.

For defining a Lie group there must exist a manifold, which at the same time possesses group properties, on which the group theory is then applied. From group axioms and the group operation the special structure of the manifold tangent vector spaces follows.

The Lie group can be presented on a trivial example: the 3D vector space:

1. It is a manifold because it can be represented in the $\mathbb{R}^3$ Euclidean space.

2. It is a group because it fulfils the group axioms, the group operation is vector addition (translation in three-space) and the identity element is the zero-vector.

Because of 1 and 2 the 3D vector space is a Lie group, but it's trivial as it is a linear space. Its tangent vector spaces coincide with the manifold itself so they have the same properties. Due to this coincidence, vectors from different tangent spaces of the manifold can be added one to another.

2.3.3 Lie algebra

On linear Lie groups the tangent space at the origin of the group is a vector space which, equipped with the matrix commutator (the already mentioned Lie bracket [,] from section 1.2.5), constitutes a real *Lie algebra* and there exists a natural correspondence between group elements and algebra elements [6].

Before going into more details and a more formal definition of the Lie algebra the definition of an algebra is presented from [6]:

> "An algebra can be constructed from a set of elements of a linear space which has a product operation, with products of the elements being elements of the set."

A real Lie algebra $o(n)$ is the vector space defined at the origin of the group and closed under the abstract Lie product [,], such that for every A, B, C of the space and $\alpha \in \mathbb{R}$ it is satisfied:

$$\begin{aligned} [A, B] &= -[B, A], \\ [\alpha A, B] &= \alpha [A, B], \\ [A + B, C] &= [A, C] + [B, C], \\ [A, [B, C]] + [B, [C, A]] + [C, [A, B]] &= 0, \end{aligned}$$

where the last equation is called the *Jacobi identity*. The first identity denotes the skew-symmetry, while the third denotes bilinearity. The Lie algebra is also closed under matrix additions and scalar multiplications.

The closure of the Lie algebra under the Lie bracket means that, if L_A and L_B are two linear vector fields, it can be shown [3]

$$[L_A, L_B] = L_C,$$

where $C = AB - BA$. This shows that the result of the Lie bracket, the commutator, belongs to the vector space defined at the origin, while the product of two skew-symmetric matrices doesn't [6]. This implies that the commutator is the operator between algebra elements, whose result is again an algebra element.

In the case of $SO(3)$ the group origin is the unitary matrix $\mathbf{I}$, so the $so(3)$ Lie algebra is defined at unity and consists of skew-symmetric matrices. What should be kept in mind about the Lie algebra is that it is a linear space. The very linearity of the algebra is what enables the use of linear numerical integrators for integration of system rotations (additions and scalar multiplications are valid).

In the section 2.3.4 exponential mapping, which is a natural parametrization of the $SO(3)$ group, is presented. The exponential mapping, along with the Lie algebra, forms the backbone of the numerical integration method presented.

2.3.4 Exponential mapping

A special orthogonal matrix $\mathbf{O}$ can be expressed as an infinite series analogous to the Taylor expansion of the exponential function exp. The matrix $\widetilde{\mathbf{n}}$ is the skew-symmetric matrix formed out of the unit vector $\vec{n}$ of the rotation axis as

$$\widetilde{\mathbf{n}} = -\epsilon_{ijk}\vec{n},$$

where ϵ_{ijk} is the *Levi-Civita permutation symbol*. The permutation creates, out of the vector $\vec{n} = \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix}^T$, the skew-symmetric matrix

$$\widetilde{\mathbf{n}} = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}. \tag{2.4}$$

If the magnitude of the rotation angle about the axis $\vec{n}$ is denoted with ψ the orientation matrix can be expressed as

$$\mathbf{O} = \exp(\psi\widetilde{\mathbf{n}}) = \exp\left(\widetilde{\boldsymbol{\psi}}\right),$$

and this expression is called *exponential mapping*. It has to be noted that special orthogonal matrices and the exp function have the same expansion, but that is where the analogy ends. For example, as generally matrices do not commute, the commutation property of the exponential function cannot be used.

The interpretation of the exponential mapping is rather simple: it can be seen as a approximation of a finite rotation by a composition of a large number of small rotations [6].

In the follow-up the exponential mapping is explained. If a curve on the Lie group, that passes through unity, is given with

$$\mathbf{A} = \mathbf{A}(t),$$
$$\mathbf{A}(0) = \mathbf{I},$$

then the tangent vector of the curve at the unity of the group has the form

$$\vec{\psi} = \left.\frac{\mathrm{d}\mathbf{A}}{\mathrm{dt}}\right|_{t=0}.$$

According to the discussions above, this tangent vector belongs to the tangent vector space at the unity of the group (manifold).

It can be shown that

$$\mathbf{A}(t) = \exp(t\mathbf{X})$$

is a solution of the initial value problem

$$\frac{\mathrm{d}\mathbf{A}}{\mathrm{dt}} = \mathbf{X}\mathbf{A}, \quad \mathbf{A}(0) = \mathbf{I}.$$

From this initial value problem and its solution it follows

$$\frac{\mathrm{d}\mathbf{A}}{\mathrm{dt}} = \frac{\mathrm{d}}{\mathrm{dt}}(\exp(t\mathbf{X})) = \mathbf{X}\exp(t\mathbf{X}) = \mathbf{X}\mathbf{A},$$

so that it can be stated that

$$\mathbf{X} = \left.\frac{\mathrm{d}\mathbf{A}}{\mathrm{dt}}\right|_{t=0}.$$

If the discussion above is transferred on the orthogonal rotation matrix, denoted now with $\mathbf{R}$, it follows that equation (2.5)

$$\mathbf{R} = \exp(t\widetilde{\boldsymbol{\omega}}) = \exp\left(\widetilde{\boldsymbol{\psi}}\right), \tag{2.5}$$

is the solution of the differential equation

$$\mathbf{R}\widetilde{\boldsymbol{\omega}} = \frac{\mathrm{d}\mathbf{R}}{\mathrm{dt}}.$$

With $\mathbf{R}(0) = \mathbf{I}$ it follows that $\widetilde{\boldsymbol{\omega}}$ is a vector (actually a matrix formed out of a vector) at $\mathbf{I}$ defined as

$$\widetilde{\boldsymbol{\omega}} = \left.\frac{\mathrm{d}\mathbf{R}}{\mathrm{dt}}\right|_{t=0}.$$

From the orthogonality property $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ the matrix $\widetilde{\boldsymbol{\omega}}$ satisfies the relation

$$\frac{\mathrm{d}\mathbf{R}\mathbf{R}^{\mathrm{T}}}{\mathrm{dt}} = \widetilde{\boldsymbol{\omega}} + \widetilde{\boldsymbol{\omega}}^T = \mathbf{0}$$

which says that $\widetilde{\boldsymbol{\omega}}$ must be skew-symmetric.

As $\mathbf{R}$ is an element of the $SO(3)$ group and $\widetilde{\boldsymbol{\omega}}$ a vector in the tangent space defined at the group unity $\mathbf{I}$, which is at the same time a Lie algebra, it follows that the exponential mapping

$$\mathbf{R} = \exp\left(t\widetilde{\boldsymbol{\omega}}\right),$$

carries vectors tangent to $SO(3)$ at $\mathbf{I}$ to elements of $SO(3)$. To put it more precisely, if the Lie algebra defined at the unity of $SO(3)$ is denoted with $so(3)$, it can be stated: the exponential map associates elements of the Lie algebra $so(3)$ to elements of the Lie group $SO(3)$: $\exp : (3) \to SO(3)$

The discussion above was intended for elements of the vector space defined at the group unity, but for elements that exist in tangent spaces at other points on the manifold a similar result is obtained. If $\mathbf{Q} = \mathbf{Q}(t)$ is a curve through $\mathbf{Q}(0) = \mathbf{R}$, then a vector tangent at $\mathbf{R}$ is given with

$$\dot{\mathbf{R}} = \left.\frac{\mathrm{d}\mathbf{Q}}{\mathrm{dt}}\right|_{t=0}.$$

The curve $\mathbf{P} = \mathbf{R}^T\mathbf{Q}(t)$ satisfies $\mathbf{P}(0) = \mathbf{I}$ and

$$\left.\frac{\mathrm{d}\mathbf{P}}{\mathrm{dt}}\right|_{t=0} = \mathbf{R}^T\dot{\mathbf{R}} = \widetilde{\boldsymbol{\omega}}, \tag{2.6}$$

which is skew-symmetric.

It turns out that a vector space tangent to $SO(3)$ at $\mathbf{R}$ (not necessarily equal to $\mathbf{I}$) is the space of matrices $\dot{\mathbf{R}}$ such that the matrix defined in equation (2.6) is skew-symmetric. The skew-symmetric matrix $\widetilde{\boldsymbol{\omega}}$ belongs to the Lie algebra defined at unity, while the matrix $\dot{\mathbf{R}}$ is not skew-symmetric and belongs to the tangent vector space defined at a point $\mathbf{R}$ on $SO(3)$. If $\dot{\mathbf{R}}$ is seen as the derivation of the rotation matrix, then $\widetilde{\boldsymbol{\omega}}$ is the mathematical representation (in the form of a skew-symmetric matrix) of the angular velocity in the Lie algebra $so(3)$ explained in the section 2.3.1.

The relation

$$\mathbf{R}\widetilde{\boldsymbol{\omega}} = \dot{\mathbf{R}} \tag{2.7}$$

is invariant under left multiplication by special orthogonal matrices, while the right invariance is valid for the relation

$$\widetilde{\boldsymbol{\omega}}'\mathbf{R} = \dot{\mathbf{R}}, \tag{2.8}$$

which is obtained when the curve $\mathbf{P}(t)$ is defined as $\mathbf{P} = \mathbf{Q}(t)\mathbf{R}^T$.

The $\widetilde{\boldsymbol{\omega}}$ from equation (2.7) in the context of rigid body dynamics is seen as the angular velocity in the body-fixed reference frame, while $\widetilde{\boldsymbol{\omega}}'$ from (2.8) is the angular velocity in the inertial reference frame. In this way, depending of the mathematical formulation, i.e. by choosing the left or right invariance, the angular velocity in different frames is obtained.

In the system governing equations (1.8) the Euler equations are of the form (1.5) and are valid only for angular velocities and angular accelerations given in the body-fixed reference frame. This way the body inertia matrix J is constant, hence, in the follow-up, the left-invariant form (2.7) is used.

Finally, for the sake of completeness, it is required to mention that the structure of the $SO(3)$ Lie group is determined by the structure of the $so(3)$ Lie algebra almost everywhere, not only close to the group unity $\mathbf{I}$ where the Lie algebra is defined.

Chapter 3

Lie group integration method for constrained MBS

In this chapter the integration method for constrained multibody systems is presented and described. In the introductory section, the previously described concepts of manifolds and Lie groups are more thoroughly described and applied to mechanical systems in a context suitable for explaining the method.

Then, the concept of local parametrization on the $SO(3)$ manifold and the possibilities it opens for integration are described. Finally, the integration algorithm is presented.

In this chapter exponential mapping is used and referred to and also other types of mappings related to the $SO(3)$ manifold are used: they are pointed out and defined when first mentioned.

3.1 The embedded manifold and parametrizations

It is known that for an unconstrained system the solution space is a k-dimensional linear vector space where $k = 6N$. On that vector space all the solutions are allowed and, as that space is linear, the only error that arises during the system integration is the error due to the integration time step length. The linear k-dimensional vector space is the Euclidean $\mathbb{R}^k$ space.

When a constraint, or a set of constraints, is imposed on the previously unconstrained system the solution space of the system changes. The system still evolves on the $\mathbb{R}^k$ space, but the imposed constraints force the system to remain on a hypersurface in $\mathbb{R}^k$ that is defined by the equations. As the constraint equations are generally non-linear, so is the resulting

hypersurface. This n-dimensional hypersurface is denoted with $\mathbb{Q}^n$ and it is mathematically defined as

$$\mathbb{M}^n = \left\{ \mathbf{q} \in \mathbb{R}^k : \mathbf{\Phi}(\mathbf{q}) = \mathbf{0} \right\}, \tag{3.1}$$

where $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$ is the set of constraints equations. Having in mind the definition of a manifold it is seen that this hypersurface is a manifold. The hypersurface $\mathbb{M}^n$ is of dimension n and it is a manifold *embedded* in $\mathbb{R}^k$, $k > n$. This coincides completely with the definition of a manifold as an entity of complex geometry that is (or can be) embedded in a space of dimension higher than the manifold dimension and which is locally representable in a chart.

Returning to the mechanical system, its solution is no longer anywhere in $\mathbb{R}^k$, but it has to lie on $\mathbb{M}^n$. As $\mathbb{M}^n$ is non-linear, the standard linear integrators (solvers) can no longer be used without corrections. Figure 3.1 shows graphically the embedded non-linear manifold $\mathbb{M}^n$ with a neighbourhood defined on it.

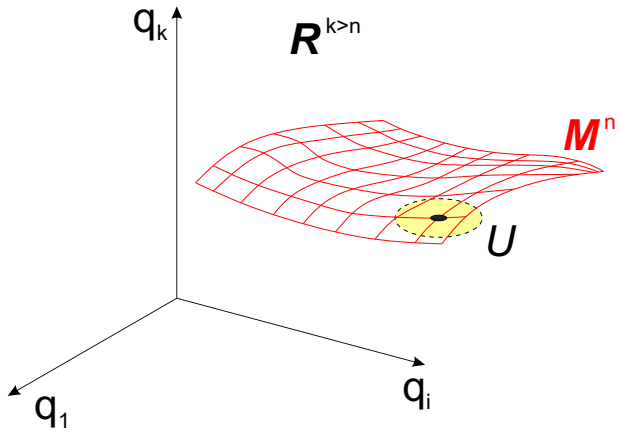


Figure 3.1: The embedded manifold $\mathbb{M}^n$ in the linear $\mathbb{R}^k$ space with a neighbourhood U on it.

In order to return back to a linear solution space the system can be re-parametrized in some neighbourhood U, where the new minimal form is then obtained. The new minimal form exists in a n-dimensional linear vector space, but is valid only locally, in the neighbourhood U. The parametrization is mathematically stated as

$$\mathbb{M}^n = \{ \mathbf{q} = \boldsymbol{\phi}(\mathbf{x}) : \mathbf{x} \in U \},$$
$$U \subset \mathbb{R}^{n=k-m},$$

where the new parameters are denoted with $\mathbf{x}$ and m is the number of constraints. From the mathematical definition above, the manifold $\mathbb{M}^n$ is defined as the collection of all the

neighbourhoods with their local parametrizations.

The original differential equation is substituted by inserting the new parameters resulting in the minimal form. In this way, linear integrators can be used and the system integrated on localized parameters in the neighbourhood U. This is the main advantage of the parametrization: linear integrators with well known properties and behaviour can be used. The drawback is that the system has to be re-parametrized for every integration step (depending on the shape of $\mathbb{M}^n$) as it changes locations (travels) on the configuration manifold $\mathbb{M}^n$.

The re-parametrization procedure is described on a simple example of an $\mathbb{R}^2$ space with an $\mathbb{M}^1$ embedded manifold. This case is shown on figure 3.2.

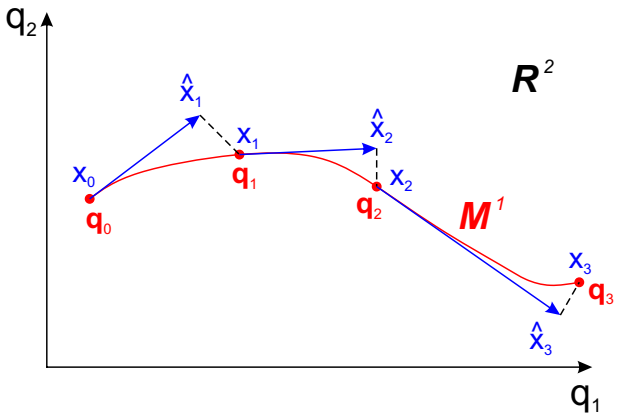


Figure 3.2: Example of the embedded manifold $\mathbb{M}^1$ in the linear $\mathbb{R}^2$ space with local parametrizations [11].

A local parametrization can be found in the form

$$\mathbf{q} = \boldsymbol{\phi}(x)\,, \tag{3.2}$$

where x is a single parameter as the dimension of the embedded manifold is equal to one. In the neighbourhood of a point of the manifold the parametrization results in a linear space without constraints on it. Using the parametrization (3.2) the original differential equation is reformulated (translated) in terms of x. As linear integrators can be used, the differential equation is integrated. The integration result (for one time step) is then obtained in terms of the new parameter x and is translated back to the manifold using the *bijection* (3.2).

In the example of figure 3.2 the local linear vector space at the manifold point $\mathbf{q}_i$ is the one-dimensional line with x_i as the local parameter. The new local position from the integration is $\widehat{x}_{i+1}$ and, after using the bijection, the point $\mathbf{q}_{i+1}$ on the manifold is obtained.

Using the same procedure, it is travelled along the manifold from that point to the next one: the local parametrization is performed first, differential equations are translated and integrated and the solution returned (transferred back) to the manifold. This way, with the solution of the differential equations, it is travelled along the manifold and at every point its local property of linearity is used. The trajectory of the system on the manifold is obtained.

The procedure described is valid generally and the local parameter x (local coordinate) can be found numerically. This was used on a general manifold, but in the sequel the local parametrization is given for the $SO\,(3)$ Lie group possessing special properties.

3.2 Local parametrization of $SO\,(3)$

Starting from the mathematical definition (2.2) of the $SO\,(3)$ manifold it is seen that it is a subgroup of $GL\,(3)$, which is non-linear and, additionally, a Lie group. In order to move from the manifold to a linear space, that allows the application of linear integrators, a parametrization has to be found and the properties of the linear space known in order to devise the complete algorithm and the bijection required.

As $SO\,(3)$ is a Lie group whose elements are 3×3 matrices denoted with $\mathbf{R}$, so will the linear vector space be a 3×3 matrix vector space. The matrix vector space $\dot{\mathbf{R}}$ is defined at every point of the manifold. A special tangent vector space was already described in section 2.3.3: the Lie algebra, a linear vector space defined at the group unity (origin) which can be denoted as $\dot{\mathbf{R}}\Big|_{\mathbf{I}}$. The algebra elements are skew-symmetric matrices.

The exponential map defined in section 2.3.4 carries elements from the Lie algebra to elements of the Lie group. As the Lie algebra is a linear space, the exponential map can be seen as a natural, group defined, parametrization of the Lie group in the neighbourhood of the Lie algebra. The exp parametrization connects a point of the manifold to an algebra element, but $\dot{\mathbf{R}}$ defines the tangent vector space at an arbitrary point of the manifold, not necessarily at unity. Because of that, the relation between the vector space at an arbitrary point of the manifold and the Lie algebra has been established in section 2.3.4. Also, it is very convenient to understand relations between vector spaces at two different arbitrary points of the manifold.

3.2.1 Correlation between vector spaces at different points on $SO(3)$

Before going into the very relation between vector spaces, the operation on the Lie group is brought back into mind. The operation on the $SO(3)$ Lie group is matrix multiplication which represents rotations composition, here defined as:

$$\mathbf{R}_{new} = \mathbf{R}_{old}\mathbf{R}, \tag{3.3}$$

where $\mathbf{R}$ is the rotation matrix that gives the change in orientation of the body between two orientations $\mathbf{R}_{old}$ and $\mathbf{R}_{new}$. It was mentioned in section 2.3.2 that the special structure and properties of tangent vector spaces of the manifold follow from the group operation. This implies that the relation between vector spaces involves the group operation, i.e. the relation is derived from the group operation.

The order of rotations in the compositions is defined with equation (3.3) because of the way rotation increments are calculated and composed within the integration method.

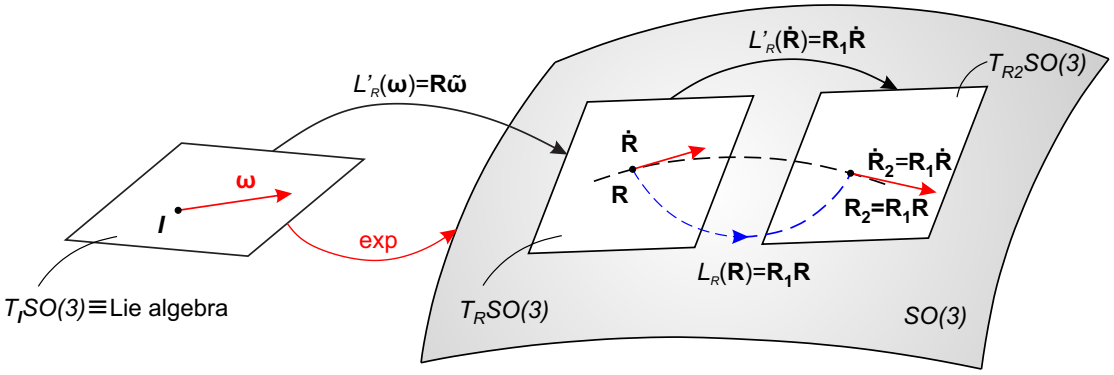


Figure 3.3: Relation between two tangent vector spaces and the Lie algebra [11].

Figure 3.3 shows the relation between two vector spaces: as two points on the Lie group are related with the group operation, while the tangent vector spaces are related with the left invariant vector field. The Lie group operation (3.3) represents translation on the $SO(3)$ group. In a similar manner the two tangent vector spaces, at points 1 and 2 of the group, are related one to another as

$$\dot{\mathbf{R}}_2 = \mathbf{R}_1\dot{\mathbf{R}}.$$

It is very important to mention that points 1 and 2 are not arbitrary and they must lie on the same integral curve on the Lie group. An integral curve on the Lie group is associated

to just one element of the Lie algebra, i.e. one angular velocity $\widetilde{\boldsymbol{\omega}}$. This is seen when the differential equation on $SO\,(3)$ is obtained from the equation above in the form

$$\dot{\mathbf{R}} = \mathbf{R}\widetilde{\boldsymbol{\omega}}. \tag{3.4}$$

Mathematical details regarding the formulation of the differential equation on $SO\,(3)$ are not presented as they are beyond the scope of this thesis. On figure 3.4 the relationship between the Lie algebra element $\boldsymbol{\omega}$ and a vector in an arbitrary point $\mathbf{R}$ tangent vector space is shown.

In the differential equation (3.4) the left invariant vector field is used. This is seen from the fact that in (3.4) the skew-symmetric matrix of the angular velocity is multiplied with the rotation matrix from the left.

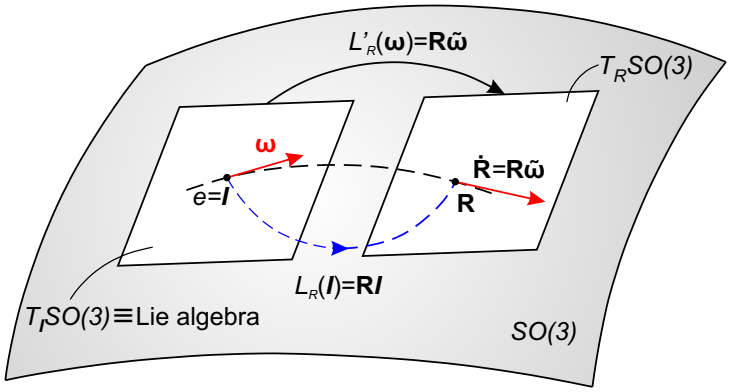


Figure 3.4: Relation between the Lie algebra and a tangent vector space [11].

3.3 Integration methods

In the discussions above it was concluded that different linear integration methods (integrators, solvers) can be used. This follows from the fact that the main problem is how to re-parametrize the $SO\,(3)$ Lie group in order to transfer the differential equation on a linear vector space. On that space virtually any linear integrator can be used. The formulation and description of the multibody system configuration space is presented first and then the algorithm that implements the *Euler integration method* is presented followed by the complete *Runge-Kutta-MK* algorithm.

3.3.1 MBS configuration space

The multibody system consists of three-dimensional bodies, so its configuration space is not purely consisting of translational configuration spaces (Euclidean $\mathbb{R}^3$ spaces) nor purely of rotation configuration spaces ($SO\,(3)$ groups). As every body possesses a position in space and a specific orientation, a single body i configuration space is composed of translational and rotational kinematic domains: $G_i = \mathbb{R}^3 \times SO\,(3)$. Thus, the complete configuration space is composed of all the bodies configuration spaces:

$$G = G_1 \times G_2 \times \ldots \times G_N = \mathbb{R}^3 \times SO\,(3) \times \mathbb{R}^3 \times SO\,(3) \times \ldots \times \mathbb{R}^3 \times SO\,(3)\,.$$

A point in the MBS configuration space is given with $p = (\mathbf{x}_1, \mathbf{R}_1, \mathbf{x}_2, \mathbf{R}_2, \ldots, \mathbf{x}_N, \mathbf{R}_N)$, and the Lie group composition operation $G \times G \to G$ is introduced by

$$\begin{aligned} p_{comp} &= p_1 \circ p_2, \\ \mathbf{x}_{comp} &= \mathbf{x}_{p_1} + \mathbf{x}_{p_2}, \\ \mathbf{R}_{comp} &= \mathbf{R}_{p_2}\mathbf{R}_{p_1}. \end{aligned}$$

Also, the identity element e is defined so that $p \circ e = e \circ p = p$, $\forall p \in G$. The identity element of the translation is the zero vector $\vec{0}$ while the identity of rotations is the unitary matrix $\mathbf{I}$. However, as the integration routines operate simultaneously on position and velocity levels, the system has to be modelled on the $2n$-dimensional Lie group, the system state-space. The group is then defined as the composition [12]

$$S = \mathbb{R}^3 \times SO\,(3) \times \ldots \times \mathbb{R}^3 \times SO\,(3) \times \ldots \times \mathbb{R}^3 \times so\,(3) \times \ldots \times \mathbb{R}^3 \times so\,(3)\,,$$

where an element of the group is given with $q = (\mathbf{x}_1, \mathbf{R}_1, \ldots, \mathbf{x}_N, \mathbf{R}_N, \dot{\mathbf{x}}_1, \widetilde{\boldsymbol{\omega}}_1, \ldots \dot{\mathbf{x}}_N, \widetilde{\boldsymbol{\omega}}_N)$. From the definition of the group element, it is also seen what the group is composed of: the first $2N$ elements belong to the Euclidean three-spaces in which the translations live and to the $SO\,(3)$ Lie groups of the rotations respectively, i.e. they form the system configuration space G. The last $2N$ elements belong again to the Euclidean three-spaces (translational velocities) and the $so\,(3)$ Lie algebras where the angular velocities exist.

Finally, the Lie algebra s and its element $z \in s$ are presented as

$$\begin{aligned} s &= \mathbb{R}^3 \times so\,(3) \times \ldots \times \mathbb{R}^3 \times so\,(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times \ldots \times \mathbb{R}^3 \times \mathbb{R}^3, \\ z &= \left(\dot{\mathbf{x}}_1, \widetilde{\boldsymbol{\omega}}_1, \ldots, \dot{\mathbf{x}}_N, \widetilde{\boldsymbol{\omega}}_N, \ddot{\mathbf{x}}_1, \dot{\widetilde{\boldsymbol{\omega}}}_1, \ldots \ddot{\mathbf{x}}_N, \dot{\widetilde{\boldsymbol{\omega}}}_N\right). \end{aligned}$$

By comparing the Lie algebra composition and a Lie group element definition, it is seen that translational velocities and accelerations, along with angular accelerations, are vector quantities while the angular velocity is a Lie algebra element: a skew-symmetric matrix.

3.3.2 MBS integration based on the Lie group Euler method

The *Euler method* or the *Euler-Cauchy method* is the most basic explicit method for numerical integration of ordinary differential equations (ODEs). It is a step-by-step method as for every time step h the same integration formula is used. The Euler method is named after the Swiss mathematician *Leonhard Euler*. The *Lie group Euler method* follows when the Euler method for linear vector spaces is applied on a Lie group. Also, this is the most simple *Munthe-Kaas method*, which, in the case of integration on Lie groups, is presented in section 3.3.3.

The vector space Euler method is a linear method whose formula follows from the *Taylor series* which gives a crude approximation of the ODE solution for a small time step. Also, this is a first order method whose global solution error is proportional to the time step length. Besides having a big error, the method also has stability issues and it's hardly ever used in practice, but it very nicely explains the methods based on Taylor series. Its simplicity makes it very suitable for introduction into numerical integration methods and it serves as the basis for more complicated methods.

If the system state-space formulation is given in general as an initial value problem

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{f}\left(t, \mathbf{z}\right), \\ \mathbf{z}\left(0\right) &= \mathbf{z}_0, \end{aligned}$$

then the system solution for the i-th integration step is based on the previous step $(i-1)$ solution, the time step length h and the simulation time t_{i-1} of the previous step:

$$\mathbf{z}_i = \mathbf{z}_{i-1} + h\mathbf{f}\left(t_{i-1}, \mathbf{z}_{i-1}\right). \tag{3.5}$$

The geometric interpretation of the method is that it is an approximation of the curve of $\mathbf{z}\left(t\right)$ by a polygon whose first side is tangent to this curve at t_0 [5]. In the sequel the solution algorithm for integration of MBS governing equations is presented. One integration step is explained in detail and the results of each operation in the step are presented. In the case of MBS integration, the Euler method is performed twice as the governing equations are of second order. The algorithm is as follows:

1. Calculate $\mathbf{F}\left(t_{i-1}, \mathbf{z}_{i-1}\right)$ using the previous step solution or, in the case of the initial step, from the initial values. The vector is calculated from the governing equation (1.8) as

$$\mathbf{F}\left(t_{i-1}, \mathbf{z}_{i-1}\right) = \begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_x^T \\ \boldsymbol{\Phi}_x & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{Q} \\ \boldsymbol{\xi} \end{bmatrix}.$$

The result consists of system accelerations and Lagrange multipliers.

2. After calculating the system vector $\mathbf{F}(t_{i-1}, \mathbf{z}_{i-1})$, separate the system accelerations which are to be integrated from the Lagrange multipliers, where the system vector is structured as

$$\mathbf{F}(t_{i-1}, \mathbf{z}_{i-1}) = \begin{bmatrix} \ddot{\mathbf{q}}_1 & \ddot{\mathbf{q}}_2 & \dots & \ddot{\mathbf{q}}_N & \lambda_1 & \lambda_2 & \dots & \lambda_K \end{bmatrix}^T = \begin{bmatrix} \ddot{\mathbf{q}} & \boldsymbol{\lambda} \end{bmatrix}^T,$$

and the vector $\ddot{\mathbf{q}}_p$, defined as $\ddot{\mathbf{q}}_p = \begin{bmatrix} \ddot{\mathbf{x}}_p & \dot{\boldsymbol{\Omega}}_p \end{bmatrix}^T$, is the p-th body acceleration vector, while $\ddot{\mathbf{q}}$ is the global system acceleration vector.

3. Integrate using the Euler method given with (3.5) to obtain the system velocities from the previous step velocities and the calculated accelerations as

$$\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_{i-1} + h\ddot{\mathbf{q}}_{i-1}.$$

4. From the obtained system velocities $\dot{\mathbf{q}}_i$, separate the translational and angular velocities as they are not integrated to displacements levels in the same way. This separation results in the system translational velocities vector $\dot{\mathbf{x}}_i$ and the system angular velocities vector $\boldsymbol{\Omega}_i$.

5. Integrate the translational velocities by using again the Euler method to obtain the translational displacements of the system:

$$\mathbf{x}_i = \mathbf{x}_{i-1} + h\dot{\mathbf{x}}_{i-1}.$$

6. Integrate the angular velocities to obtain the system orientations. As the orientations are given with the rotation matrices $\mathbf{R}^p$ for each body p, $p = \{1, \dots, N\}$, independently so the new rotational matrices have to be calculated for each body separately. As none of the local parametrizations for $SO(3)$ is used and, by using the Lie group Euler, the system is integrated directly on $SO(3)$:

$$\mathbf{R}_i^p = \mathbf{R}_{i-1}^p \exp\left(h\boldsymbol{\Omega}_{i-1}^p\right).$$

7. Run steps 1 to 6 with a fixed integration time step length h until the simulation end time T_{end} is reached.

3.3.3 MBS integration based on the MK integration method

The *Runge-Kutta-Munthe-Kaas* (RK-MK, usually only MK) integration method is the classical *Runge-Kutta* (RK) integration method applied for the integration on Lie groups. The MK method originally uses the exponential map as the mapping function between the Lie algebra and the Lie group, but also the Cayley map can be used for the same operation.

The RK method is a whole family methods (distinguished by the method order) of great practical importance and much greater accuracy than the above described Euler method. The method was developed by the German mathematicians *Karl Runge* and *Wilhelm Kutta*.

In each integration step of the RK method auxiliary quantities are calculated. Their number and way of calculation depends on the method order. The most commonly used RK method is of fourth order and is often referred to as RK4 or *classical Runge-Kutta method*. RK methods are numerically stable and the fourth-order method's global error is proportional to the fourth power of the integration time step length: $O(h^4)$. Furthermore, RK methods can be formulated as implicit or explicit iterative methods and it's a one-step method meaning that in each step only data from the preceding step is used.

Returning to the integration of MBS, the input of the MK method, same ad for the Lie group Euler method, consists of the function of the (differential) governing equations, system initial conditions, integration time step length and the simulation end time (or number of steps).

In order to apply the method on a Lie group, in each integration step a parametrization is introduced in order to move from the non-linear Lie group to the linear Lie algebra space. As the Lie algebra is a linear vector space, the linear RK method can be applied there and the solution returned back to the group (using a mapping function). In this way the MK method is obtained.

The discussion above is applied in the case of the $SO(3)$ Lie group. Using the dexp^{-1} map, the equations are "pushed-forward" from the $SO(3)$ group to the $so(3)$ algebra, the numerical integration is then performed and the result is "pulled-back", using the exp map, from the algebra to the new position on the $SO(3)$ group. The group structure is respected in every integration step, as in each step the system is re-parametrized and the equations transferred in the Lie algebra. In the Lie algebra the equations are integrated and the solution returned to the last known point on the group (the preceding integration step result).

In the sequel the integration algorithm for the MB system solution with the MK4 method (MK of order four) implemented is presented step by step. In the fourth order method for

integration on $SO(3)$ the $dexp^{-1}$ mapping arises and the way of calculating it is given together with the formulae for the exponential and Cayley maps in section 3.5.

The MK integration method of order four, for the solution of MBS motion, is performed in the following steps:

1. From the previous step solution, or from the initial conditions values in the case of the first integration step, calculate $\mathbf{F}(t_{i-1}, \mathbf{z}_{i-1})$ using

$$\mathbf{F}(t_{i-1}, \mathbf{z}_{i-1}) = \begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_x^T \\ \mathbf{\Phi}_x & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{Q} \\ \boldsymbol{\xi} \end{bmatrix}.$$

2. After calculating the system vector $\mathbf{F}(t_{i-1}, \mathbf{z}_{i-1})$, separate the system accelerations from the Lagrange multipliers. The system vector is structured as

$$\mathbf{F}(t_{i-1}, \mathbf{z}_{i-1}) = \begin{bmatrix} \ddot{\mathbf{q}}^1 & \ddot{\mathbf{q}}^2 & \dots & \ddot{\mathbf{q}}^N & \lambda^1 & \lambda^2 & \dots & \lambda^K \end{bmatrix}^T = \begin{bmatrix} \ddot{\mathbf{q}}_1 & \boldsymbol{\lambda}_1 \end{bmatrix}^T,$$

and the vector $\ddot{\mathbf{q}}_1^p$, defined as $\ddot{\mathbf{q}}_1^p = \begin{bmatrix} \ddot{\mathbf{x}}_1^p & \dot{\mathbf{\Omega}}_1^p \end{bmatrix}^T$, is the p-th body acceleration vector, while $\ddot{\mathbf{q}}_1$ is the global system acceleration vector. This accelerations, together with the previous step solution, form the first set of auxiliary quantities of the MK method. This is denoted with the index 1.

3. Calculate the 2^{nd} set of auxiliary quantities as

$$\begin{aligned} \mathbf{a}_2^p &= \mathbf{x}_{i-1}^p + \frac{h}{2}\dot{\mathbf{x}}_{i-1}^p, \\ \mathbf{b}_2^p &= \dot{\mathbf{x}}_{i-1}^p + \frac{h}{2}\ddot{\mathbf{x}}_1^p, \\ \mathbf{c}_2^p &= \mathbf{\Omega}_{i-1}^p + \frac{h}{2}\dot{\mathbf{\Omega}}_1^p, \\ \mathbf{u}_2^p &= \frac{h}{2}\mathbf{\Omega}_1^p, \\ \mathbf{d}_2^p &= \mathbf{R}_{i-1}^p \exp(\mathbf{u}_1^p), \end{aligned}$$

and from these values calculate the auxiliary quantities used in for the calculation of the next set of auxiliary quantities:

$$\begin{aligned} &\mathbf{F}\left(\frac{h}{2} + t_{i-1}, \mathbf{a}_2, \mathbf{b}_2, \mathbf{c}_2, \mathbf{d}_2\right), \\ &\mathbf{k}_{2R}^p = \mathrm{dexp}^{-1}(\mathbf{c}_2^{\mathrm{p}}). \end{aligned}$$

Again, from $\mathbf{F}\left(\frac{h}{2} + t_{i-1}, \mathbf{a}_2, \mathbf{b}_2, \mathbf{c}_2, \mathbf{d}_2\right)$ accelerations ($\ddot{\mathbf{x}}_2^p$ and $\dot{\mathbf{\Omega}}_2^p$) are separated from the Lagrange multipliers. It is seen that $\mathbf{F}$ is calculated using the 1^{st} set of auxiliary quantities.

4. Calculate the 3^{rd} set of auxiliary quantities as

$$\begin{aligned}
\mathbf{a}_3^p &= \mathbf{x}_{i-1}^p + \frac{h}{2}\mathbf{b}_2^p, \\
\mathbf{b}_3^p &= \dot{\mathbf{x}}_{i-1}^p + \frac{h}{2}\ddot{\mathbf{x}}_2^p, \\
\mathbf{c}_3^p &= \mathbf{\Omega}_{i-1}^p + \frac{h}{2}\dot{\mathbf{\Omega}}_2^p, \\
\mathbf{u}_3^p &= \frac{h}{2}\mathbf{k}_{2R}^p, \\
\mathbf{d}_3^p &= \mathbf{R}_{i-1}^p \exp\left(\mathbf{u}_3^p\right),
\end{aligned}$$

and from these values calculate the auxiliary quantities used in for the calculation of the next set of auxiliary quantities:

$$\begin{aligned}
&\mathbf{F}\left(\frac{h}{2} + t_{i-1}, \mathbf{a}_3, \mathbf{b}_3, \mathbf{c}_3, \mathbf{d}_3\right), \\
&\mathbf{k}_{3R}^p = \mathrm{dexp}^{-1}\left(\mathbf{c}_3^{\mathrm{p}}\right).
\end{aligned}$$

From $\mathbf{F}\left(\frac{h}{2} + t_{i-1}, \mathbf{a}_3, \mathbf{b}_3, \mathbf{c}_3, \mathbf{d}_3\right)$, accelerations $\ddot{\mathbf{x}}_3^p$ and $\dot{\mathbf{\Omega}}_3^p$ are separated.

5. Calculate the 4^{th} set of auxiliary quantities as

$$\begin{aligned}
\mathbf{a}_4^p &= \mathbf{x}_{i-1}^p + h\mathbf{b}_3^p, \\
\mathbf{b}_4^p &= \dot{\mathbf{x}}_{i-1}^p + h\ddot{\mathbf{x}}_3^p, \\
\mathbf{c}_4^p &= \mathbf{\Omega}_{i-1}^p + h\dot{\mathbf{\Omega}}_3^p, \\
\mathbf{u}_4^p &= h\mathbf{k}_{3R}^p, \\
\mathbf{d}_4^p &= \mathbf{R}_{i-1}^p \exp\left(\mathbf{u}_4^p\right),
\end{aligned}$$

and from these values calculate the last set of auxiliary quantities:

$$\begin{aligned}
&\mathbf{F}\left(h + t_{i-1}, \mathbf{a}_4, \mathbf{b}_4, \mathbf{c}_4, \mathbf{d}_4\right), \\
&\mathbf{k}_{4R}^p = \mathrm{dexp}^{-1}\left(\mathbf{c}_4^{\mathrm{p}}\right).
\end{aligned}$$

Again, separate accelerations $\ddot{\mathbf{x}}_4^p$ and $\dot{\mathbf{\Omega}}_4^p$ from the vector $\mathbf{F}\left(\frac{h}{2} + t_{i-1}, \mathbf{a}_4, \mathbf{b}_4, \mathbf{c}_4, \mathbf{d}_4\right)$.

6. Calculate the final result for translational displacements and velocities and angular velocities of the current integration step as

$$\begin{aligned}
\mathbf{x}_i^p &= \mathbf{x}_{i-1}^p + \frac{h}{6}\left(\dot{\mathbf{x}}_{i-1}^p + 2\mathbf{b}_2^p + 2\mathbf{b}_3^p + \mathbf{b}_4^p\right), \\
\dot{\mathbf{x}}_i^p &= \dot{\mathbf{x}}_{i-1}^p + \frac{h}{6}\left(\ddot{\mathbf{x}}_1^p + 2\ddot{\mathbf{x}}_2^p + 2\ddot{\mathbf{x}}_3^p + \ddot{\mathbf{x}}_4^p\right), \\
\mathbf{\Omega}_i^p &= \mathbf{\Omega}_{i-1}^p + \frac{h}{6}\left(\dot{\mathbf{\Omega}}_1^p + 2\dot{\mathbf{\Omega}}_2^p + 2\dot{\mathbf{\Omega}}_3^p + \dot{\mathbf{\Omega}}_4^p\right).
\end{aligned}$$

7. Calculate the auxiliary quantities required for the calculation of the current integration step rotation matrices:

$$\boldsymbol{\phi}_R^p = \frac{h}{6}\left(\boldsymbol{\Omega}_{i-1}^p + 2\mathbf{c}_2^p + 2\mathbf{c}_3^p + \mathbf{c}_4^p\right).$$

8. Calculate the current integration step rotation matrices using the exponential map as

$$\mathbf{R}_i^p = \mathbf{R}_{i-1}^p \exp\left(\boldsymbol{\phi}_R^p\right).$$

9. Repeat steps 1 to 8 with a fixed integration time step length h until the simulation end time T_{end} is reached.

With this, the two methods used for solving the motion of the case study MBS (chapter 4) are presented. In the following section the problem of constraint violation is shortly presented and the constraint violation stabilisation procedure, that is implemented in order to correct the integrator results, is given. Also, at the end of the chapter, expressions for calculating different mappings are given and shortly described.

3.4 Constraint violation stabilisation algorithm

As the system is numerically integrated and the constraint equations in (1.8) are at the acceleration level, it is expected that, in the joints of the system, errors arise at the velocity and displacement levels. The acceleration level is automatically satisfied as the constraints acceleration level equations are incorporated in the system governing equations formulation, but, due to the straightforward integration procedure, errors at velocities and displacements accumulate.

The error (inability) of the integrator solution to satisfy joints and imposed motions constraints at the displacement and velocity level is called *constraint violation*. This is often referred to as *drift* from the configuration manifold of the system, as the integrator solution in that case is no longer on the system configuration manifold. The constraint violation at the displacement level is easily calculated if one joint (that connects the first and second body, for example) global position is calculated using the first body global position, its rotation matrix and the joint position in its local reference frame, and then the same is calculated from the second body parameters. The difference between the joint position calculated from the first body of the joint and from the second body of the joint gives the error in the joint position.

The position error at the joint described above implies that constraints are violated at the displacement level. The same can be shown for the velocity level also. The system displacements and velocities have to be stabilised, so that the integrator displacement and velocity solutions are corrected. Also, the constraint violation can influence the dynamics of the system significantly (see the case study results presentation in section 4.5).

In order to eliminate the constraint violation, i.e. reduce it to an acceptable level which does not affect the system dynamics and its solution, the the intermediate integrator results have to be stabilised, both at the velocity and displacement levels. With the stabilisation, that is performed in each integration step, the system solution is kept on the system configuration manifold guaranteeing the correct solution.

The implemented stabilisation procedure is carried out using global integration coordinates in each integration time step. The integrator output in each step forms the input of the stabilisation algorithm. Using the least squares method, the algorithm iterates to the correct displacement and velocity values, that satisfy the constraint equations at the velocity and displacement levels. The global system constraints equations at the displacement and velocity levels are of the form

$$\mathbf{\Phi}\left(\mathbf{x}, \mathbf{R}\right) = \mathbf{0}, \tag{3.6}$$

$$\dot{\mathbf{\Phi}}\left(\mathbf{x}, \mathbf{R}, \dot{\mathbf{x}}, \mathbf{\Omega}\right) = \mathbf{0}. \tag{3.7}$$

The stabilisation algorithm implementation is as follows:

1. After the integration procedure is carried out for the current integration step (denoted with the index i), the integrator output, consisting of the displacements $\widehat{\mathbf{x}}_i^p$, rotation matrices $\widehat{\mathbf{R}}_i^p$ and translational and angular velocities $\widehat{\dot{\mathbf{x}}}_i^p$ and $\widehat{\dot{\mathbf{\Omega}}}_i^p$, forms the input of the stabilisation procedure. This solution follows from the step 6 of the MBS integration based the Lie group Euler algorithm or from the step 8 of the MBS integration based on the MK method.

2. Stabilise the system displacements by solving the non-linear least squares problem

$$\left(\mathbf{\Phi}\left(\widehat{\mathbf{x}}_i, \widehat{\mathbf{R}}_i\right) - \mathbf{\Phi}\left(\mathbf{x}_i, \mathbf{R}_i\right)\right)^2 \rightarrow \min,$$
$$\left(\mathbf{R}_i^p \, \mathbf{R}_i^{p\,T} - \mathbf{I}\right)^2 \rightarrow \min.$$

   It is seen from the problem formulation that the square of the error between the integrator solution and the stabilised solution (i.e. the constraint equations values

obtained from those solutions) is minimized. Also, one additional constraint that has to be satisfied by the stabilisation procedure is the orthogonality of rotation matrices. The rotation matrices obtained from the integrator are orthogonal (the mappings always result in orthogonal matrices), but when the rotations and displacements are stabilised this orthogonality can be lost, so the additional constraint is required.

3. The solution of the least squares problem for the displacement level are the stabilised displacements $\mathbf{x}_i^p$ and stabilised rotation matrices $\mathbf{R}_i^p$ of the system. Together with $\widehat{\dot{\mathbf{x}}}_i^p$ and $\widehat{\dot{\mathbf{\Omega}}}_i^p$, the stabilised displacements form the input into the stabilisation of system velocities.

4. Stabilise the system velocities (the system stabilised displacements and rotation matrices from the stabilisation step 2 are fixed) again by solving a least squares problem given in the form
$$\left(\dot{\mathbf{\Phi}}\left(\widehat{\dot{\mathbf{x}}}_i, \widehat{\dot{\mathbf{\Omega}}}_i\right) - \dot{\mathbf{\Phi}}\left(\dot{\mathbf{x}}_i, \dot{\mathbf{\Omega}}_i\right)\right)^2 \rightarrow \min .$$

5. Gather the stabilised integration step results $\dot{\mathbf{x}}_i$, $\mathbf{x}_i$, $\mathbf{R}_i$ and $\mathbf{\Omega}_i$ for all system bodies ($i = \{1, \ldots, N\}$). They are the stabilised system solutions of the current integration step and present the input for the next integration step. These stabilised solutions lie on the configuration manifold, in contrary to the intermediate integrator results which drifted from the configuration manifold.

The stabilisation steps 1 to 5 are repeated at each integration step, because at each step the integrator intermediate result drifts from the manifold. Non-stabilised solutions can lead to errors in the system dynamics (forces and torques act on different points than in the real system due to drift) and, consequently, to wrong solutions.

Other types of stabilisations also exist, but the review and implementation of other stabilisation methods is beyond the scope of the thesis.

3.5 Mappings calculation

In the previous sections, exponential mapping was often mentioned, its properties and interpretation discussed, but the way the mapping is practically computed was not given. In this section the way different mappings are numerically computed is presented. For the numerical computation of mappings, as they are used in possibly large scale systems, it is

very important that they are fast and stable. Of course, the computations must yield a required accuracy and stability level, so that it makes sense to use them in practice.

Mappings used in the thesis include the, already mentioned, exponential mapping and the Cayley mapping. The Cayley map is another mapping function between the $so\,(3)$ Lie algebra and the $SO\,(3)$ Lie group. The exponential map is treated first and, afterwards, the Cayley mapping is presented. The expressions for the mappings calculation are taken from [3].

3.5.1 Exponential mapping calculation

The exponential map was already thoroughly described throughout the thesis, but the way the matrices are numerically computed was not mentioned. Using that map, from a skew-symmetric matrix $\widetilde{\mathbf{a}}$ an orthogonal matrix $\mathbf{B} \in \mathbb{R}^{3\times 3} : \mathbf{B}\mathbf{B}^T = \mathbf{I}$ is obtained, i.e. an element of the Lie algebra is mapped to an element of the Lie group: $so\,(3) \to SO\,(3)$. The skew-symmetric matrix $\widetilde{\mathbf{a}}$ is obtained from the vector $\mathbf{a}$, from which the vector norm (vector length) is computed as

$$\varphi = \|\mathbf{a}\| = \sqrt{\mathbf{a}^T\mathbf{a}}. \tag{3.8}$$

The exponential map is then computed using

$$\exp\left(\widetilde{\mathbf{a}}\right) = \mathbf{I} + \frac{\sin\varphi}{\varphi}\widetilde{\mathbf{a}} + \frac{\sin^2\left(\varphi/2\right)}{2\left(\varphi/2\right)^2}\widetilde{\mathbf{a}}^2, \tag{3.9}$$

and this is the well known *Rodrigues formula* for the calculation of the exponential map in the case of the $SO\,(3)$ Lie group. The exponential mapping of a matrix cannot be expressed exactly with a finite number of operations with elementary functions. This follows from the fact that computers cannot calculate exactly the trigonometric functions: for their calculations computers use the Taylor series expansion and approximate the function. It is important that the approximate solution of the map belongs to $SO\,(3)$ within machine accuracy [2].

The differential and the inverse of the differential exponential map are calculated using

$$\operatorname{dexp}\left(\widetilde{\mathbf{a}}\right) = \frac{\exp\left(\widetilde{\mathbf{a}}\right) - \mathbf{I}}{\widetilde{\mathbf{a}}} = \mathbf{I} + \frac{\sin^2\left(\varphi/2\right)}{\varphi^2/2}\widetilde{\mathbf{a}} + \frac{\varphi - \sin\varphi}{\varphi^3}\widetilde{\mathbf{a}}^2, \tag{3.10}$$

$$\operatorname{dexp}^{-1}\left(\widetilde{\mathbf{a}}\right) = \frac{\widetilde{\mathbf{a}}}{\exp\left(\widetilde{\mathbf{a}}\right) - \mathbf{I}} = \mathbf{I} - \frac{1}{2}\widetilde{\mathbf{a}} - \frac{\varphi\cot\left(\varphi/2\right) - 2}{2\varphi^2}\widetilde{\mathbf{a}}^2. \tag{3.11}$$

3.5.2 Cayley mapping calculation

Another transformation that maps a special orthogonal matrix to a skew-symmetric matrix is the Cayley transformation originally described and named after the British mathematician *Arthur Cayley*. The mapping is applicable on the $SO(3)$ Lie group and its $so(3)$ Lie algebra, but in the Cayley map singularities arise. Because of that, the map is not generally applicable to big rotational displacements that go beyond the map domain. The Cayley map is calculated from

$$\mathrm{cay}\left(\widetilde{\mathbf{a}}\right) = \mathbf{I} + \mathrm{c}\widetilde{\mathbf{a}} + \frac{\mathrm{c}}{2}\widetilde{\mathbf{a}}^2, \tag{3.12}$$

where the auxiliary quantity c is calculated from the vector norm φ of equation (3.8) as

$$c = \frac{4}{4 + \varphi^2}.$$

The Cayley map of equation (3.12) is numerically faster and easier to compute than the exponential map of (3.9) and could be also used to provide faster algorithms and computations of body spatial rotations (as long as the mapping singularity is not encountered). This is inspected in the case study presented in chapter 4.

The differential of the Cayley mapping and the inverse of the differential are calculated using

$$\mathrm{dcay}\left(\widetilde{\mathbf{a}}\right) = c\left(\mathbf{I} + \frac{1}{2}\widetilde{\mathbf{a}}\right), \tag{3.13}$$

$$\mathrm{dcay}^{-1}\left(\widetilde{\mathbf{a}}\right) = \mathbf{I} - \frac{1}{2}\widetilde{\mathbf{a}} + \frac{1}{4}\mathbf{a}\mathbf{a}^T. \tag{3.14}$$

All the expressions for the computation of mappings are given and, in section 4.5, the difference in the computation times between the given mappings is compared.

Chapter 4

Case study

In this chapter the described integration methods are used for integrating i.e. solving the motion of a spatial multibody system. Firstly, the MBS is described: bodies and joints are identified, as are the forces that act on the system. Also, the prescribed motions in the system are defined along with the system initial configuration and velocities. Three different motion cases with different imposed motions and external forces are defined. After that introductory part of the case study, the kinematic constraint equations are formulated and reshaped in a form suitable for integration. After the system governing equations and methods are finally described, the system is implemented in *MATLAB* and the solution obtained using two different mapping operators from the Lie algebra to the Lie group: the exponential map and the Cayley map. The solutions are compared with the *ADAMS* solution of the system motion.

4.1 Multibody system description

The system under consideration is a cylindrically shaped satellite equipped with a manipulator consisting of three members connected with different joints. The system is presented on figure 4.1 where it is seen that it is a four body spatial system.

The first manipulator element, the base rod, is connected with the satellite body via a spherical joint. The base rod is then connected with a revolute joint to the second manipulator element, the slider rod, on which there is a tool, the slider, connected with a prismatic joint to it and able to move along the slider rod.

The bodies in the system are numbered from 1 to 4 and on figure 4.1 the body-fixed reference frames ($x_i y_i z_i$, $i = \{1, \ldots, 4\}$) are shown along with the inertial reference frame

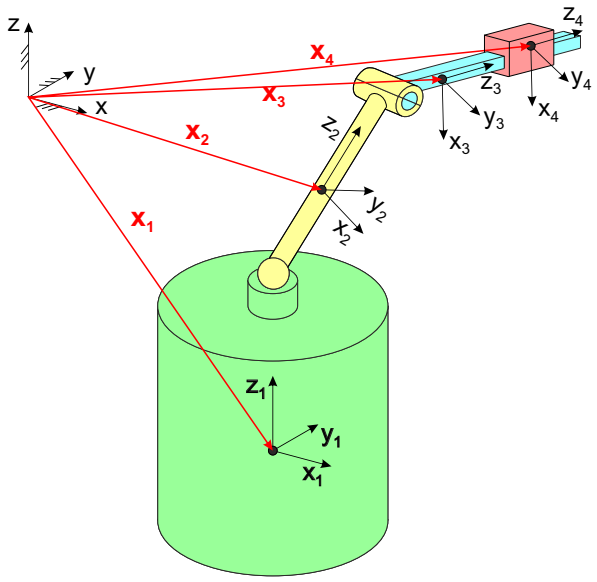


Figure 4.1: The four body MBS analysed in the case study.

xyz.

The system presented here is a simplified satellite system. More complex systems can also be defined with different inertial and geometric properties, but the scope of the thesis and this case study is to show the possibilities of the Lie group integration algorithms, not on the complexity of satellites and their manipulators.

4.1.1 Bodies parameters

In this section the parameters of individual bodies in the system are identified. The necessary parameters are identified and they include:

- body dimensions and geometrical shape description,
- mass of the body, and
- locations of joints on the body expressed in the body-fixed reference frame.

The conventions and simplifications used here are:

- the body-fixed reference frame is located in the body centre of gravity;
- bodies are regarded as homogeneous with uniform mass distribution;

- simplified body representations are used;
- the joints geometry (construction) is not taken into account.

In accordance with the statements above, the body inertia matrix is formed and the mass moments of inertia determined. Also, the system is discretized using primitive, or as simple as possible, geometrical bodies so that the inertia can be calculated in the simplest possible manner. Furthermore, all bodies used are symmetric and their body-fixed reference frame axes are at the same time the principal axes of inertia. The inertia matrix (for the principal axes of inertia) of the i-th body is defined as

$$\mathbf{J}_i = \begin{bmatrix} J_x^i & 0 & 0 \\ 0 & J_y^i & 0 \\ 0 & 0 & J_z^i \end{bmatrix},$$

where J_k^i, $k = \{x_i, y_i z_i\}$, is the moment of inertia about the k-axis calculated using formulae given in [1]. The above defined inertia matrix is a diagonal matrix and consequently J_k^i are principal mass moments of inertia.

Main satellite body

The main satellite body is a cylindrically shaped object and it is modelled as a pure cylinder shown on figure 4.2. The body-fixed reference frame $x_1 y_1 z_1$ is also shown along with the body dimensions d_1, L_1, the spherical joint location vector $\mathbf{X}_1^{SJ}$ and the fixed point locations in the body-fixed reference frame and in the inertial reference frame (vectors $\mathbf{X}_1^{FP}$ and $\mathbf{x}^{FP}$).

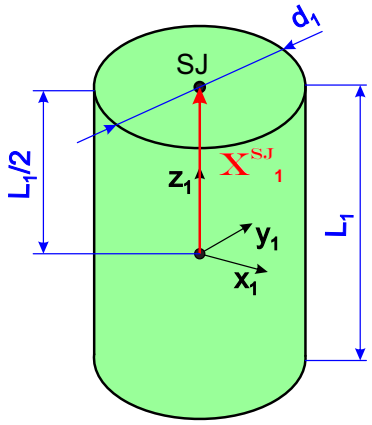


Figure 4.2: The main satellite body: dimensions and joint position vector.

$$d_1 = 2\,\text{m},$$
$$L_1 = 3\,\text{m},$$
$$m_1 = 1800\,\text{kg},$$

$$\mathbf{X}_1^{SJ} = \begin{bmatrix} 0 & 0 & \frac{L_1}{2} \end{bmatrix}^T,$$
$$\mathbf{X}_1^{FP} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T,$$
$$\mathbf{x}^{FP} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T.$$

From the quantities defined above it is possible to calculate the body moments of inertia about the principal axes of inertia, which coincide with the body-fixed reference frame axes:

$$J_x^1 = J_y^1 = \frac{m_1 d_1^2}{16} + \frac{m_1 L_1^2}{12},$$
$$J_z^1 = \frac{m_1 d_1^2}{8}.$$

The body inertia matrix is formed using the values obtained from the expressions above.

Base rod

The base rod is the first manipulator element of the satellite manipulator and it is denoted as body 2. It's connected to the satellite body with a spherical joint whose location on the body is given with the vector $\mathbf{X}_2^{SJ}$ and with the second manipulator element, the slider rod, is connected with a revolute joint whose location is given with $\mathbf{X}_2^{RJ}$ (figure 4.3). Geometrically, the base rod is modelled as a long cylindrical rod.

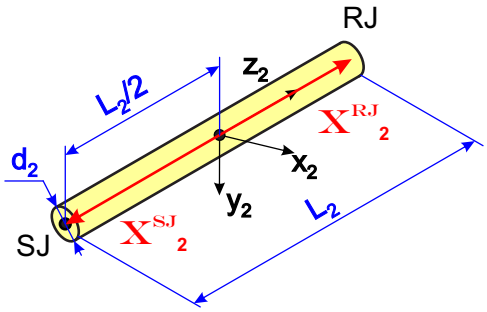


Figure 4.3: The base rod: dimensions and joint position vectors.

$$d_2 = 0.15 \text{ m},$$
$$L_2 = 1.5 \text{ m},$$
$$m_2 = 80 \text{ kg},$$

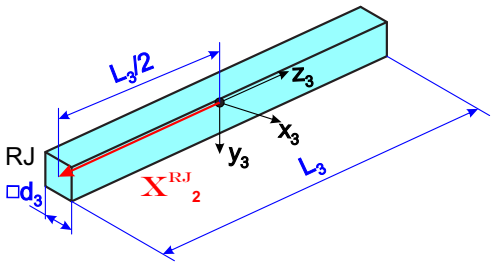
$$\mathbf{X}_2^{SJ} = \begin{bmatrix} 0 & 0 & -\frac{L_2}{2} \end{bmatrix}^T,$$
$$\mathbf{X}_2^{RJ} = \begin{bmatrix} 0 & 0 & \frac{L_2}{2} \end{bmatrix}^T.$$

The moments of inertia are calculated as

$$J_x^2 = J_y^2 = \frac{m_2 d_2^2}{16} + \frac{m_2 L_2^2}{12},$$
$$J_z^2 = \frac{m_2 d_2^2}{8}.$$

Slider rod

The slider rod is, in contrary to the base rod, modelled as a long rectangular prism with a small cross section. As on the slider rod the slider is allowed only to slide along it, with that geometrical shape all relative rotations between the rod and slider are constrained.



$$d_3 = 0.1\ \text{m},$$
$$L_3 = 1.5\ \text{m},$$
$$m_3 = 60\ \text{kg},$$

Figure 4.4: The slider rod: dimensions and joint position vector.

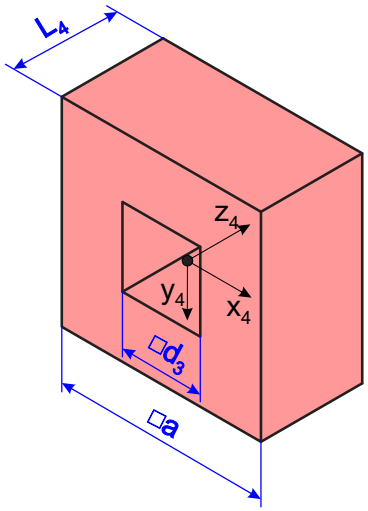
$$\mathbf{X}_3^{RJ} = \begin{bmatrix} 0 & 0 & -\frac{L_3}{2} \end{bmatrix}^T.$$

The slider rod moments of inertia are

$$J_x^3 = J_y^3 = \frac{m_3\left(d_3^2 + L_3^2\right)}{12},$$
$$J_z^3 = \frac{m_3 d_3^2}{6}.$$

Slider

As it was mentioned above, the slider translates along the slider rod and is modelled as a rectangular prism with a rectangular hole in the centre. In the case of the satellite manipulator, on the slider a tool or gripper for maintenance of the satellite or docking other objects that approach the satellite can be mounted.



$$d_3 = 0.1\ \text{m},$$
$$a = 0.3\ \text{m},$$
$$L_4 = 0.2\ \text{m},$$
$$m_4 = 30\ \text{kg},$$

Figure 4.5: The slider: dimensions.

In order to calculate the slider moments of inertia, additional parameters are defined as

$$\beta' = \frac{V'}{V} = \frac{bd_3}{a^2 - d_3^2}, \qquad \beta'' = \frac{V''}{V} = \frac{ba}{a^2 - d_3^2},$$
$$m' = \beta' m_4, \qquad m'' = \beta'' m_4,$$
$$b = \frac{a - d_3}{2},$$

and they follow from the decomposition of the slider into simpler parts, whose inertia is easy to calculate.

The coefficients β' and β'' follow when the volumes of sections of the slider are divided with the volume of the complete slider. As the slider has uniform mass distribution, that ratio is used to obtain the mass of individual sections. The slider moments of inertia are calculated from the inertias of smaller parts of the slider by using the *Huygens-Steiner theorem* (parallel axis theorem). It finally follows:

$$J_x^4 = J_y^4 = \frac{1}{6} m' \left[L_4^2 + b^2 + 3\left(b + d_3\right)^2 \right] + \frac{1}{6} m'' \left(L_4^2 + a^2 \right),$$
$$J_z^4 = \frac{1}{6} m' \left[b^2 + d_3^2 + 3\left(b + d_3\right)^2 \right] + \frac{1}{6} m'' \left[b^2 + a^2 + 3\left(b + d_3\right)^2 \right].$$

Now, as all parameters for the system topological definition, along with inertial properties, are given, the system is fully defined and it remains to define the external forces that act on the system and its initial conditions for displacements and velocities.

4.1.2 System initial conditions

As the system governing equations are differential equations of the second order (forming, together with the algebraic constraint equations, a DAE system of equations), for each coordinate involved two initial conditions have to be given in order to have a starting point for obtaining a trajectory on the solution manifold. The initial conditions are given for the displacements (system initial configuration) while the velocities are calculated.

In the case study, three different motion cases of the system are analysed. The initial displacements of the system are given first and are valid for all three motions cases. The motion cases differ in the angular velocities and number of the imposed motions and in external forces. The description and parameters of the motion cases analysed in the case study is given in section 4.1.3. Also, in that section the way initial velocities are calculated is described.

Initial displacements

In its initial configuration the satellite manipulator lies in the global xz plane. The cylindrical satellite body local coordinate system is in the initial position coincident with the global coordinate system. Mathematically this is expressed as

$$\mathbf{x}_1^0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T,$$
$$\boldsymbol{\theta}_1^0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T,$$

where $\boldsymbol{\theta}_1^0$ is the initial vector of the Euler angles for the 313 rotation sequence from which, by using equation (2.3), the initial body orientation matrix is calculated to be

$$\mathbf{R}_1^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The base rod of the satellite manipulator has the same local coordinate system orientation as the global coordinate system and the main satellite body. The CoG location of body 2 in the global coordinate system and its orientation is given with

$$\mathbf{x}_2^0 = \begin{bmatrix} 0 & 0 & \frac{L_1+L_2}{2} \end{bmatrix}^T,$$
$$\boldsymbol{\theta}_2^0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T,$$
$$\mathbf{R}_2^0 = \mathbf{R}_1^0.$$

The slider rod and the slider have the same orientation as their relative rotations are constrained (defined and explained in section 4.2.4). In the initial position, the slider rod stands vertically as a continuation of the base rod. Mathematically, the initial position is given with

$$\mathbf{x}_3^0 = \begin{bmatrix} 0 & 0 & \frac{L_1}{2} + L_2 + \frac{L_3}{2} \end{bmatrix}^T,$$
$$\boldsymbol{\theta}_3^0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T,$$
$$\mathbf{R}_3^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In the initial position, besides having the same orientation as the slider rod, the slider is located right in the middle of the slider rod so that their global positions (of their centres of

gravity) coincide in the initial configuration:

$$\mathbf{x}_4^0 = \mathbf{x}_3^0,$$
$$\boldsymbol{\theta}_4^0 = \boldsymbol{\theta}_3^0,$$
$$\mathbf{R}_4^0 = \mathbf{R}_3^0.$$

With this all the initial conditions regarding positions are given and this information is also used for determining the velocity initial conditions.

4.1.3 Motion cases

As was stated above, the motion cases analysed differ in the velocities of imposed motions, their number and external forces that act on the system bodies. The external forces that act on the system are presented first, followed by the description of the imposed motions velocities. In the following subsections, all the variants for external forces and velocities are given. The motion cases are finally described and their parameters defined as a combination of external forces and imposed motions velocities.

Initial velocities

The system initial velocities are calculated using the system constraints defined in section 4.2. The constraint equations, used for calculating the system initial velocities, have to be formulated at the velocity level, so that from the values of the system initial displacement and the imposed motions velocities are obtained. In this section the system velocities that remain constant throughout the motion, i.e. the quantities that define the rheonomic constraints (imposed motions), are presented.

The i-th body (or element) angular velocity expressed in the global coordinate system is denoted with $\boldsymbol{\Omega}_i^G$ and the angular and translational velocity vectors are given as:

$$\boldsymbol{\Omega}_i = \begin{bmatrix} \Omega_i^x & \Omega_i^y & \Omega_i^z \end{bmatrix}^T, \qquad \dot{\mathbf{x}}_i = \begin{bmatrix} \dot{x}_i^x & \dot{x}_i^y & \dot{x}_i^z \end{bmatrix}^T.$$

The multibody system is solved for three different motion cases, but all the cases are formed using two variants of imposed velocities. The first velocity variant is a simpler, 1 DOF rotational motion imposed on the satellite body and the spherical joint, while the second is a complex, 3 DOF rotation. The superscript index, next to the global coordinate system index G, on the imposed motions constant velocity denotes the motion variant identifier.

The main satellite body is set to have a constant angular velocity. The two variants for the angular velocity are expressed in terms of the inertial reference frame (global coordinate system) as

$$\boldsymbol{\Omega}_1^{G,1} = \begin{bmatrix} 0 & 0 & \frac{2\pi}{60} \end{bmatrix}^T \text{ rad/s}, \qquad \boldsymbol{\Omega}_1^{G,2} = \begin{bmatrix} 0 & \frac{2\pi}{60} & \frac{3\pi}{60} \end{bmatrix}^T \text{ rad/s}. \tag{4.1}$$

Between the main satellite body and the first manipulator element - the base rod, there is a spherical joint that does not transmit rotations between the two bodies. The base rod rotation is also a prescribed motion in the system: the vector of the base rod angular velocity is constant in the global reference frame and it is equal to

$$\boldsymbol{\Omega}_2^{G,1} = \begin{bmatrix} \frac{\pi}{120} & 0 & 0 \end{bmatrix}^T \text{ rad/s}, \qquad \boldsymbol{\Omega}_2^{G,2} = \begin{bmatrix} \frac{\pi}{120} & \frac{2\pi}{60} & 0 \end{bmatrix}^T \text{ rad/s} \tag{4.2}$$

The revolute joint motion results from the solution of the system dynamics, but the slider has a prescribed relative motion with respect to the slider rod. The initial velocity of the slider, relative to the slider rod and along the z_i translation axes, equals to

$$\dot{x}_{3,4}^{rel} = 0.02 \,\text{m/s}, \tag{4.3}$$

while the constant acceleration on the slider, also along the z_i axes is equal to

$$\ddot{x}_{3,4}^{rel} = -0.0007 \,\text{m/s}^2. \tag{4.4}$$

In two cases the slider motion is as prescribed, but in a third case its motion is not included and a force acting along the slider translational axis is included so that the system then has two dynamic DOFs.

Finally, the revolute joint motion follows from the forces that act on the system. The satellite manipulator is controlled with the moment acting on the revolute joint. The actuator moment is in relation with the force acting on the slider and the slider imposed motion. The joint is dynamically controlled so that its relative angular acceleration is constant and equals

$$\dot{\Omega}_x^{RJ} = -0.00058 \,\text{rad/s}^2. \tag{4.5}$$

This acceleration is imposed around the only axis about which the revolute joint allows rotations (the x_2 and x_3 axes).

In order to calculate the system initial velocities, the joint constraints together with the rheonomic constraints equations have to be formulated at the velocity level and solved for translational and angular velocities. In the section 4.2 the constraint equations are

formulated, joints are defined and the imposed motions mathematically modelled. The quantities given with (4.1) to (4.5) are constants that appear in the rheonomic constraints mathematical formulation of section 4.2.5.

The system initial velocities are calculated using a numerical algorithm: the velocity level constraint equations (4.43) are solved numerically using the least squares method and the solution obtained are the initial velocities. The input of the global system constraint equations at the velocity level consists of the system constant velocities given above and the system initial positions (section 4.1.2).

External forces

The external applied system forces are forces that don't arise due to the interaction of the bodies that form the system (e.g. constraint forces, internal forces). They include forces like gravity, forces due to contacts with other bodies, forces originating from the propulsion system, etc.

On the satellite, gravity is not acting as it can be considered that it is in a gravity-free environment. Consequently, the only applied forces that act are the actuator forces and external loads. In the formulated system, external forces act on the slider and the slider rod. On the slider the force (the first variant) is a constant load due to the tool mounted and it is defined in the local coordinate system of the slider along the y_4 axis. On the slider rod the applied force is a time-dependent moment of the joint actuator acting along the joint x_3 axis. The external forces are defined in the respective body-fixed reference frames as

$$\mathbf{F}_4^1 = \begin{bmatrix} F_{x_4} \\ F_{y_4} \\ F_{z_4} \end{bmatrix} = \begin{bmatrix} 0 \\ 25 \\ 0 \end{bmatrix} \text{ N}, \qquad \mathbf{L}_3 = \begin{bmatrix} L_{x_3} \\ L_{y_3} \\ L_{z_3} \end{bmatrix} = \begin{bmatrix} L_{x_3}(t) \\ 0 \\ 0 \end{bmatrix} \text{ Nm}.$$

The second variant of the forces is valid, as it is seen later, when there is no imposed motion on the slider. In that case, the force acting on the slider is collinear with the slider translation axis and it is defined as

$$\mathbf{F}_4^2 = \begin{bmatrix} F_{x_4} \\ F_{y_4} \\ F_{z_4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -0.014 \end{bmatrix} \text{ N}.$$

In the second variant of external loads, the revolute joint actuator moment remains unchanged.

The time-dependent moment of the actuator is calculated in such a way that the resulting acceleration of the slider rod is constant. The time-dependence of the moment follows from the imposed motion on the slider on which the force acts. The moment time-function is

$$L_{x_3}(t) = J_x^3 \dot{\Omega}_x^{RJ} + F_{y_4}\left(x_{3,4}^{rel}(t) + \frac{L_3}{2}\right), \tag{4.6}$$

where the time-function of the relative slider motion is denoted with $x_{3,4}^{rel}(t)$ (for more details see section 4.2.5), and $\dot{\Omega}_x^{RJ}$ is the constant revoulte joint acceleration.

As the revolute joint actuator acts simultaneously on the base rod and the slider rod, it is necessary to include the same moment of opposite sign on the base rod as

$$\mathbf{L}_{x_2} = -\mathbf{L}_{x_3}.$$

The moments are, like the angular velocities, expressed in the Euler equation in terms of the body-fixed reference frame so they are introduced in the global formulation as they are given here, but the forces have to be expressed in the inertial reference frame. The slider force is constant in the body-fixed reference frame, but in the inertial frame it is expressed as

$$\mathbf{F}_4^G = \mathbf{R}_4 \mathbf{F}_4,$$

and as such it is introduced in the mathematical formulation.

These forces are added to the non-linear velocity forces in the global forces vector $\mathbf{Q}$, and in that way they are mathematically introduced into the governing equations given with (1.8). Also, as no motion is imposed on the revolute joint between the slider rod and the base rod, these forces, together with inertia forces, determine the resulting relative motion of the revolute joint.

Motion cases formulation

The motion cases analysed in the case study are formulated using the imposed motions and external forces defined in the subsections above. The analysed motion cases of the system are:

Motion case 1 The main satellite body has a 1 DOF rotation about its local z_1 axis, which is coincident with the global z axis throughout the motion. The motion constant angular velocity is given with $\mathbf{\Omega}_1^{G,1}$, defined in (4.1). The base manipulator rod motion is also a 1 DOF rotation about the global x axis with constant angular velocity $\mathbf{\Omega}_2^{G,1}$ given with (4.2). The external force on the slider is given with $\mathbf{F}_4^1$. The motion of the

system is interpreted as: the main satellite body rotates about its longitudinal axis, while the satellite manipulator is loaded (the slider tool carries an external object). The satellite manipulator motion is controlled so that motions are imposed on the spherical joint and the slider but there is one dynamical DOF in the revolute joint. The revolute joint motion is force-controlled, so that it results in a constant angular acceleration of the revolute joint.

Motion case 2 The main satellite body and the base rod of the satellite manipulator have the same imposed motions as in the motion case 1: $\mathbf{\Omega}_1^{G,1}$ and $\mathbf{\Omega}_2^{G,1}$. The difference is in the number of dynamic DOFs and in the slider motion. The slider is not controlled kinematically, as in this case its actuators are failing. At the same time the force on the slider changes to $\mathbf{F}_4^2$, so that there is just a small force acting in the slider translation z_4 axis direction: this implies that the slider motion is now a dynamical DOF together with the revoulte joint motion. The revolute joint motion is controlled dynamically with the moment of the joint actuator. In this motion case the system possesses one dynamical DOF and one rheonomic constraint less than the other two motion cases (the slider imposed motion is not included).

Motion case 3 This motion case is a showcase that the method is able to solve systems with large rotations that would result in singularities if a local parametrization (ex. Euler angles) had been adopted. The rotational motion of the main satellite body and the base rod consist of 3 DOF rotations given with $\mathbf{\Omega}_1^{G,2}$ and $\mathbf{\Omega}_2^{G,2}$. The system has one dynamical DOF and the force on the slider is given with $\mathbf{F}_4^1$, as was the case in motion case 1. The slider relative translational motion is also imposed.

4.2 Constraint equations formulation

As it is stated in section 1.1.3, the constraint equations have to be reshaped to the acceleration level. The governing equations in the DAE index 1 form, which are suitable for integration using the methods presented in chapter 3, are then obtained. The starting point for the formulation of constraints at the acceleration level is the formulation of constraints at the displacement level. Depending on the constraint type the constraint equations can be formulated at the velocity level, or even at the acceleration level.

From the constraint equations at the displacement level by differentiating twice with respect to time, the constraint equations at the acceleration level are obtained (for holonomic

systems). This approach is used in the sequel where the constraint equations of four basic joints are formulated: the fixed point constraint, the spherical, revolute and prismatic joints are mathematically formulated.

Before going into the very formulation of the joints constraint equations some relations, that are used throughout the equations formulation, are presented. Firstly, the difference between the local and global coordinate system is to be made. The local or body-fixed coordinate system is a non-inertial reference frame fixed to the body. The global coordinate system is the inertial reference frame in which the Newton 2^{nd} law is valid. Also, the system motion is naturally given in terms of the global coordinate system as the local coordinate system of a body cannot describe its motion, i.e. it moves with the body.

If the position of a point P in the i-th body local coordinate system is denoted with $\mathbf{X}_P^i$ and the same point is denoted in the global coordinate system with $\mathbf{x}_P^i$, then the relation between the two coordinate systems is given with equation (4.7):

$$\mathbf{x}_P^i = \mathbf{R}_i \mathbf{X}_P^i. \tag{4.7}$$

In the sequel, the superscript i denoting the body is omitted. As the point P is a point fixed on the body (does not move relatively to the body) it follows

$$\dot{\mathbf{X}}_P = \frac{\mathrm{d}\mathbf{X}_\mathrm{P}}{\mathrm{dt}} = 0. \tag{4.8}$$

Differentiating (4.7) with respect to time and using (4.8) it is obtained

$$\dot{\mathbf{x}}_P = \dot{\mathbf{R}}\mathbf{X}_P.$$

The time derivative of the rotation matrix $\mathbf{R}$ is defined as

$$\dot{\mathbf{R}} = \mathbf{R}\widetilde{\mathbf{\Omega}}, \tag{4.9}$$

where $\widetilde{\mathbf{\Omega}}$ is the skew-symmetric matrix of the body angular velocity expressed in the body-fixed reference frame. This results in the application of left invariant vector field. After using (4.9) it follows:

$$\dot{\mathbf{x}}_P = \mathbf{R}\widetilde{\mathbf{\Omega}}\mathbf{X}_P. \tag{4.10}$$

Relations (4.7) to (4.10) are often used when formulating the constraint equations, especially when the equations are differentiated to obtain the acceleration level equations. Also, some other properties that belong to the vector cross product and to the vector scalar product are used.

The vector cross product can be calculated, besides using the symbolic determinant, using the skew-symmetric matrix representation of a vector (given with equation (2.4)):

$$\vec{\Omega} \times \vec{X}_P = \widetilde{\mathbf{\Omega}}\mathbf{X}_P.$$

The anti-commutativity property of the vector cross product yields

$$\begin{aligned} \vec{\Omega} \times \vec{X}_P &= -\vec{X}_P \times \vec{\Omega}, \\ \widetilde{\mathbf{\Omega}}\mathbf{X}_P &= -\widetilde{\mathbf{X}}_P\mathbf{\Omega}. \end{aligned} \tag{4.11}$$

The vector scalar product is calculated using the vector transpose:

$$\vec{x} \cdot \vec{y} = \mathbf{x}^T\mathbf{y},$$

and it is commutative so that

$$\mathbf{x}^T\mathbf{y} = \mathbf{y}^T\mathbf{x}. \tag{4.12}$$

Finally, unit vectors, that give the coordinate axes orientation, are also often used so they are defined at this point:

$$\mathbf{e}^x = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad \mathbf{e}^y = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad \mathbf{e}^z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

As such, they can be used either in the global coordinate system or in local, body-fixed reference frames. If they are expressed in the i-th body local coordinate system then they are expressed in the global coordinate system as

$$\mathbf{E}_i^x = \mathbf{R}_i\mathbf{e}^x, \qquad \mathbf{E}_i^y = \mathbf{R}_i\mathbf{e}^y, \qquad \mathbf{E}_i^z = \mathbf{R}_i\mathbf{e}^z. \tag{4.13}$$

These are some mathematical formalisms that are used throughout this section. Other properties of vectors and their products are also used when formulating the initial constraint equations at the displacements level, but they are explained at the specific joint formulation for whose formulation the property is used.

4.2.1 Fixed point constraint

The fixed point constraint is one of the simplest constraint that eliminates DOFs of a body. This constraints force a body point to have a fixed position in the inertial reference

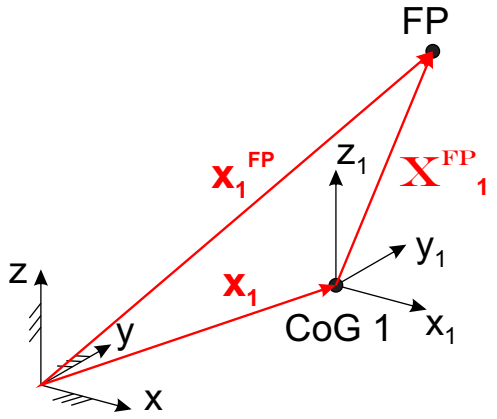


Figure 4.6: Fixed point position vectors.

frame throughout the motion. Consequently, this leads to the elimination of body translational DOFs, while rotations about that fixed point are allowed. The number of eliminated DOFs is equal to three.

The mathematical formulation of such a constraint follows from the analysis of vectors from figure 4.6. If the body fixed point position is expressed in the body-fixed reference frame with $\mathbf{X}_1^{FP}$ and that point is constrained to have the position $\mathbf{x}_1^{FP}$ in the inertial reference frame the mathematical formulation states

$$\mathbf{\Phi}^{FP} = \mathbf{x}_1^{FP} - \mathbf{x}_1 - \mathbf{R}_1\mathbf{X}_1^{FP} = \mathbf{0}. \tag{4.14}$$

After differentiating with respect to time and using relations (4.9), (4.11) and the fact that $\mathbf{x}_1^{FP}$ is a fixed point in space (independent of time) the velocity level is obtained in the form

$$\dot{\mathbf{\Phi}}^{FP} = -\dot{\mathbf{x}}_1 - \mathbf{R}_1\widetilde{\mathbf{\Omega}}_1\mathbf{X}_1^{FP} = \mathbf{0}. \tag{4.15}$$

Differentiating once more with respect to time and performing some more mathematical manipulations the acceleration level constraint equation is obtained in the form

$$-\ddot{\mathbf{x}}_1 + \mathbf{R}_1\widetilde{\mathbf{X}}_1^{FP}\dot{\mathbf{\Omega}}_1 = -\mathbf{R}_1\widetilde{\mathbf{\Omega}}_1\widetilde{\mathbf{X}}_1^{FP}\mathbf{\Omega}_1.$$

Written in the form

$$\mathbf{\Phi}_x^{FP}\ddot{\mathbf{x}}_1 = \boldsymbol{\xi}^{FP}, \tag{4.16}$$

the matrix $\mathbf{\Phi}_x^{FP}$ and vector $\boldsymbol{\xi}^{FP}$ are

$$\mathbf{\Phi}_x^{FP} = \begin{bmatrix} -\mathbf{I} & +\mathbf{R}_1\widetilde{\mathbf{X}}_1^{FP} \end{bmatrix},$$
$$\boldsymbol{\xi}^{FP} = -\mathbf{R}_1\widetilde{\mathbf{\Omega}}_1\widetilde{\mathbf{X}}_1^{FP}\mathbf{\Omega}_1,$$

where the acceleration vector $\ddot{\mathbf{x}}_1$ in equation (4.16) contains both translational and angular accelerations of body 1.

It is finally seen that the matrix equation (4.16) consists of three algebraic equations for three eliminated DOFs and, for the matrix dimension to be correct, the unitary matrix $\mathbf{I}$ dimension is $\dim(\mathbf{I}) = 3 \times 3$. The position of the fixed body point in the inertial reference frame is not directly seen at the acceleration level, but it's given in the initial conditions with the body CoG initial position $\mathbf{x}_1^0$ and the position of the fixed point $\mathbf{X}_1^{FP}$ in the body-fixed reference frame.

In the next section the spherical joint mathematical formulation is presented, and it is seen that the fixed point and the spherical joint constraints give practically the same equations. The only difference is that the fixed point given here is with respect to a point fixed in the inertial reference frame. This is why in the final constraint equation of the fixed point constraint only one body accelerations and parameters are seen.

4.2.2 Spherical joint

The spherical joint is graphically shown on figure 4.7. This kind of joint doesn't place any restrictions on the relative rotational displacements between the two bodies connected with that joint, but restricts bodies relative displacements.

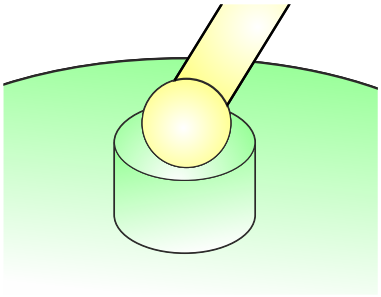


Figure 4.7: Spherical joint.

The bodies connected with the joint are the main satellite body (body 1) and the base rod of the manipulator (body 2). On figure 4.8 the location vectors of the joint in the local and global coordinate systems are shown. Also, the coordinate systems involved are shown for an arbitrary relative position of the two bodies.

On figure 4.8 with CoG is denoted the body centre of gravity, while SJ stands for spherical joint. The joint is mathematically modelled so that the spherical joint global position is the same when calculated from both bodies CoG position. In that formulation, bodies global

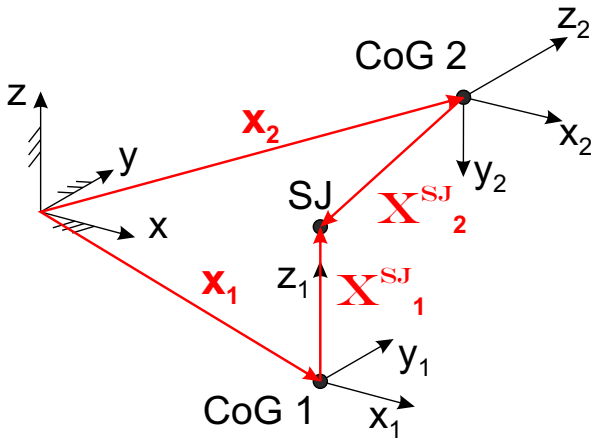


Figure 4.8: Spherical joint position vectors.

CoG positions, rotation matrices and joint local positions are involved (equation (4.17)). One point on the first body is restricted to have the same global position as one point on the second body and that point is the spherical joint location. Finally, three algebraic constraint equations at the displacement level are obtained, written in matrix form they are

$$\mathbf{\Phi}^{SJ} = \mathbf{x}_1 + \mathbf{R}_1\mathbf{X}_1^{SJ} - \mathbf{x}_2 - \mathbf{R}_2\mathbf{X}_2^{SJ} = \mathbf{0}. \tag{4.17}$$

Vectors $\mathbf{X}_1^{SJ}$ and $\mathbf{X}_2^{SJ}$ give the location of the spherical joint in the body coordinate system while $\mathbf{x}_1$ and $\mathbf{x}_2$ are the positions of the bodies CoGs. Differentiating (4.17) once with respect to time the constraint is obtained at the velocity level:

$$\dot{\mathbf{\Phi}}^{SJ} = \dot{\mathbf{x}}_1 + \mathbf{R}_1\widetilde{\mathbf{\Omega}}_1\mathbf{X}_1^{SJ} - \dot{\mathbf{x}}_2 - \mathbf{R}_2\widetilde{\mathbf{\Omega}}_2\mathbf{X}_2^{SJ} = \mathbf{0}. \tag{4.18}$$

Differentiating once more and separating the terms dependent on accelerations on the left-hand side and other terms on the right-hand side it is obtained

$$\ddot{\mathbf{x}}_1 - \ddot{\mathbf{x}}_2 - \mathbf{R}_1\widetilde{\mathbf{X}}_1^{SJ}\dot{\mathbf{\Omega}}_1 + \mathbf{R}_2\widetilde{\mathbf{X}}_2^{SJ}\dot{\mathbf{\Omega}}_2 = \mathbf{R}_2\widetilde{\mathbf{\Omega}}_2\widetilde{\mathbf{\Omega}}_2\mathbf{X}_2^{SJ} - \mathbf{R}_1\widetilde{\mathbf{\Omega}}_1\widetilde{\mathbf{\Omega}}_1\mathbf{X}_1^{SJ}.$$

The spherical joint constraint equations in the matrix form, analogous to the second equation of (1.8), are

$$\mathbf{\Phi}_x^{SJ}\ddot{\mathbf{x}}_{1,2} = \boldsymbol{\xi}^{SJ}, \tag{4.19}$$

where $\ddot{\mathbf{x}}_{1,2} = \begin{bmatrix}\ddot{\mathbf{x}}_1 & \dot{\mathbf{\Omega}}_1 & \ddot{\mathbf{x}}_2 & \dot{\mathbf{\Omega}}_1\end{bmatrix}^T$ is the vector of accelerations of bodies 1 and 2 that are connected with the spherical joint. The spherical constraint matrix $\mathbf{\Phi}_x^{SJ}$ and vector $\boldsymbol{\xi}^{SJ}$ are

defined as

$$\mathbf{\Phi}_x^{SJ} = \begin{bmatrix} \mathbf{I} & -\mathbf{R}_1\widetilde{\mathbf{X}}_1^{SJ} & -\mathbf{I} & \mathbf{R}_2\widetilde{\mathbf{X}}_2^{SJ} \end{bmatrix},$$
$$\boldsymbol{\xi}^{SJ} = \mathbf{R}_2\widetilde{\mathbf{\Omega}}_2\widetilde{\mathbf{\Omega}}_2\mathbf{X}_2^{SJ} - \mathbf{R}_1\widetilde{\mathbf{\Omega}}_1\widetilde{\mathbf{\Omega}}_1\mathbf{X}_1^{SJ}.$$

Analysing equation (4.19) and its terms it is seen that, as $\dim(\mathbf{R}_i) = \dim\left(\widetilde{\mathbf{\Omega}}_i\right) = \dim\left(\widetilde{\mathbf{X}}_i^{SJ}\right) = 3 \times 3$ and $\dim\left(\mathbf{X}_i^{SJ}\right) = 3 \times 1$, this matrix equation consists of three equations implying that between the two bodies connected with the spherical joint three DOFs are eliminated. The eliminated DOFs are the relative translations at the joint location, while relative body rotations are still allowed. Also, from the dimensions of matrices, it follows that the unitary matrix $\mathbf{I}$ dimension is $\dim(\mathbf{I}) = 3 \times 3$.

4.2.3 Revolute joint

The second joint of the satellite manipulator is the revolute joint that connects the base rod (body 2) with the slider rod (body 3). This kind of joint prevents relative body translations while, at the same time, allowing only one relative rotational displacement. Figure 4.9 shows the revolute joint graphically.

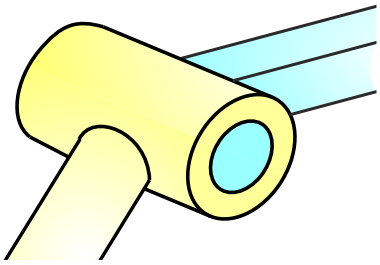


Figure 4.9: Revolute joint.

On figure 4.10 the revolute joint (RJ) position vectors are shown along with the coordinate axes direction vectors that are also used for the constraint equations formulation.

The first step of mathematically defining the revolute joint is to eliminate the relative translations at the joint location. This is done in completely the same way the spherical joint was formulated. Consequently, the first set of constraint equations of the revolute joint are the same equations as the spherical joint equations, but with indices modified so that bodies 2 and 3 are brought into relation:

$$\mathbf{\Phi}_1^{RJ} = \mathbf{x}_2 + \mathbf{R}_2\mathbf{X}_2^{RJ} - \mathbf{x}_3 - \mathbf{R}_3\mathbf{X}_3^{RJ} = \mathbf{0}. \tag{4.20}$$

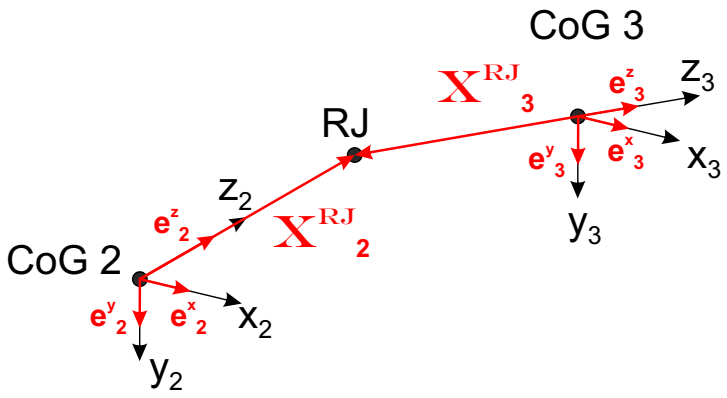


Figure 4.10: Revolute joint position vectors.

The first set of constraints at the velocity level are then of the form

$$\dot{\mathbf{\Phi}}_1^{RJ} = \dot{\mathbf{x}}_2 + \mathbf{R}_2\widetilde{\mathbf{\Omega}}_2\mathbf{X}_2^{RJ} - \dot{\mathbf{x}}_3 - \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{X}_3^{RJ} = \mathbf{0}, \tag{4.21}$$

while at the acceleration level they are

$$\mathbf{\Phi}_{x,1}^{RJ}\ddot{\mathbf{x}}_{2,3} = \boldsymbol{\xi}_1^{RJ}, \tag{4.22}$$

where

$$\mathbf{\Phi}_{x,1}^{RJ} = \begin{bmatrix} \mathbf{I} & -\mathbf{R}_2\widetilde{\mathbf{X}}_2^{RJ} & -\mathbf{I} & \mathbf{R}_3\widetilde{\mathbf{X}}_3^{RJ} \end{bmatrix},$$
$$\boldsymbol{\xi}_1^{RJ} = \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\widetilde{\mathbf{\Omega}}_3\mathbf{X}_3^{RJ} - \mathbf{R}_2\widetilde{\mathbf{\Omega}}_2\widetilde{\mathbf{\Omega}}_2\mathbf{X}_2^{RJ}.$$

With the matrix equation (4.22) in place, relative translations are eliminated, but there are still two relative rotations that have to be constrained. This means that two additional scalar equations have to be formulated.

Only the rotation about the local x_2 and x_3 axes is allowed. This implies that the planes y_2z_2 and y_3z_3 have to be parallel at all times and this parallelism is achieved by eliminating relative rotations about the y_i and z_i axes. Mathematically, this is stated using the scalar product and its geometrical interpretation. The scalar product of unit vectors in the x_3 and z_2 direction and the scalar product of unit vectors in the x_2 and y_3 direction are set to be zero:

$$\mathbf{\Phi}_2^{RJ} = \mathbf{E}_3^{z^T}\mathbf{E}_2^x = 0,$$
$$\mathbf{\Phi}_3^{RJ} = \mathbf{E}_3^{y^T}\mathbf{E}_2^x = 0,$$

where the vectors $\mathbf{E}$ are defined with relations (4.13) and, when inserted, it is obtained

$$\begin{aligned} \mathbf{\Phi}_2^{RJ} &= \left(\mathbf{R}_3\mathbf{e}^z\right)^T \mathbf{R}_2\mathbf{e}^x = 0, \\ \mathbf{\Phi}_3^{RJ} &= \left(\mathbf{R}_3\mathbf{e}^y\right)^T \mathbf{R}_2\mathbf{e}^x = 0. \end{aligned} \tag{4.23}$$

This follows from the vector scalar product geometrical interpretation: if two vectors are perpendicular their scalar product is zero. What equations (4.23) state is:

1. The local coordinate axis z_3 of body 3 is perpendicular to the local coordinate axis x_2 of body 2. This constraint eliminates relative rotations about the y_i axes.

2. The local coordinate axis y_3 of body 3 is perpendicular to the local coordinate axis x_2 of body 2. This constraint eliminates relative rotations about the z_i axes.

These statements can be easily verified by imagining the relative rotations of the two bodies local reference frames and then performing the scalar product.

Having formulated the two rotational constraints, it is necessary to reshape them to the acceleration level. After the first differentiation with respect to time equations (4.24) follow:

$$\begin{aligned} \dot{\mathbf{\Phi}}_2^{RJ} &= \left(\mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^z\right)^T \mathbf{R}_2\mathbf{e}^x + \left(\mathbf{R}_3\mathbf{e}^z\right)^T \mathbf{R}_2\widetilde{\mathbf{\Omega}}_2\mathbf{e}^x = 0, \\ \dot{\mathbf{\Phi}}_3^{RJ} &= \left(\mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^y\right)^T \mathbf{R}_2\mathbf{e}^x + \left(\mathbf{R}_3\mathbf{e}^y\right)^T \mathbf{R}_2\widetilde{\mathbf{\Omega}}_2\mathbf{e}^x = 0. \end{aligned} \tag{4.24}$$

After differentiating once more with respect to time and performing some mathematical manipulations, using the vector scalar product commutativity of equation (4.12), the general form of the constraints at the acceleration level for the two additional rotational constraints of the revolute joint are obtained

$$\mathbf{\Phi}_{x,2}^{RJ}\ddot{\mathbf{x}}_{2,3} = \boldsymbol{\xi}_2^{RJ}. \tag{4.25}$$

The matrix $\mathbf{\Phi}_{x,2}^{RJ}$ and vector $\boldsymbol{\xi}_2^{RJ}$ are defined as

$$\mathbf{\Phi}_{x,2}^{RJ} = \begin{bmatrix} \mathbf{0} & -\left(\mathbf{R}_3\mathbf{e}^z\right)^T \mathbf{R}_2\widetilde{\mathbf{e}}^x & \mathbf{0} & -\left(\mathbf{R}_2\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{e}}^z \\ \mathbf{0} & -\left(\mathbf{R}_3\mathbf{e}^y\right)^T \mathbf{R}_2\widetilde{\mathbf{e}}^x & \mathbf{0} & -\left(\mathbf{R}_2\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{e}}^y \end{bmatrix},$$

$$\boldsymbol{\xi}_2^{RJ} = \begin{bmatrix} \left[\left(\mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^z\right)^T \mathbf{R}_2\widetilde{\mathbf{e}}^x + \left(\mathbf{R}_3\mathbf{e}^z\right)^T \mathbf{R}_2\widetilde{\mathbf{\Omega}}_2\widetilde{\mathbf{e}}^x\right]\mathbf{\Omega}_2 \\ \qquad + \left[\left(\mathbf{R}_2\widetilde{\mathbf{\Omega}}_2\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{e}}^z + \left(\mathbf{R}_2\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\widetilde{\mathbf{e}}^z\right]\mathbf{\Omega}_3 \\ \left[\left(\mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^y\right)^T \mathbf{R}_2\widetilde{\mathbf{e}}^x + \left(\mathbf{R}_3\mathbf{e}^y\right)^T \mathbf{R}_2\widetilde{\mathbf{\Omega}}_2\widetilde{\mathbf{e}}^x\right]\mathbf{\Omega}_2 \\ \qquad + \left[\left(\mathbf{R}_2\widetilde{\mathbf{\Omega}}_2\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{e}}^y + \left(\mathbf{R}_2\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\widetilde{\mathbf{e}}^y\right]\mathbf{\Omega}_3 \end{bmatrix}.$$

Since the matrices given above consist of only two rows (this can be deduced when looking at the dimension of individual elements and from the knowledge that these are two equations), the null matrix $\mathbf{0}$ is of dimension $\dim(\mathbf{0}) = 1 \times 3$.

The complete constraint equations in matrix form of the revolute joint are finally formulated by simply putting the two matrix equations (4.22) and (4.25) into one single matrix equation

$$\mathbf{\Phi}_x^{RJ} \ddot{\mathbf{x}}_{2,3} = \boldsymbol{\xi}^{RJ}, \tag{4.26}$$

where the revolute joint constraint matrix and vector are

$$\mathbf{\Phi}_x^{RJ} = \begin{bmatrix} \mathbf{\Phi}_{x,1}^{RJ} \\ \mathbf{\Phi}_{x,2}^{RJ} \end{bmatrix}, \qquad \boldsymbol{\xi}^{RJ} = \begin{bmatrix} \boldsymbol{\xi}_1^{RJ} \\ \boldsymbol{\xi}_2^{RJ} \end{bmatrix}.$$

From the equations given above, it is seen that there are five scalar equations. Consequently, five relative DOFs of the two bodies connected with the revolute joint are eliminated. The only remaining DOF is one rotation about the x_2 and x_3 axes, which are collinear at all times.

4.2.4 Prismatic joint

A prismatic joint is a sliding contact joint where just one relative translational motion is allowed between the two connected bodies. Figure 4.11 shows the prismatic joint that connects the slider rod (body 3) and the slider (body 4).

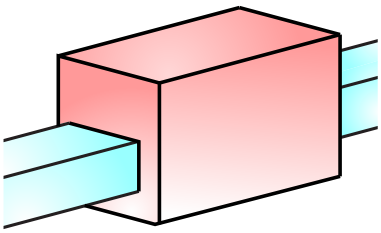


Figure 4.11: Prismatic joint.

Figure 4.12 shows the coordinate systems of the bodies connected with the prismatic joint (PJ) and all the vectors that are used when formulating the constraint equations of the joint.

When formulating the constraint equations of the prismatic joint, the first step is to eliminate all the relative rotations between the two connected bodies as the joint allows only

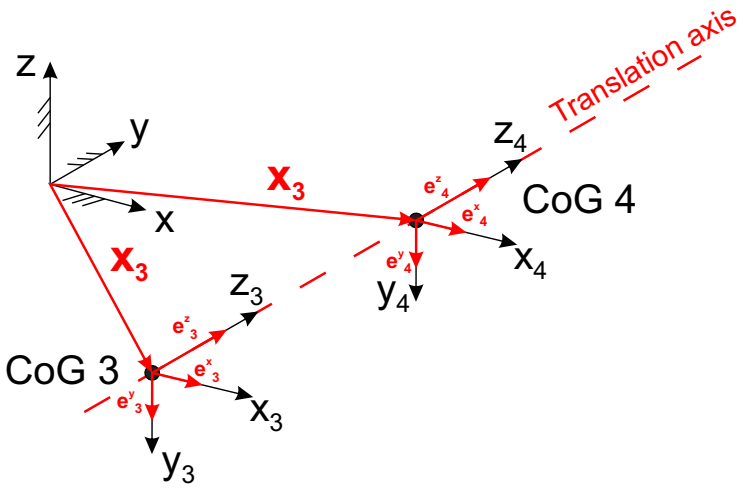


Figure 4.12: Prismatic joint position vectors.

one translation. The rotations are constrained in the same way rotations were constrained in the revolute joint formulation of section 4.2.3.

In order to constrain all relative rotations using the vector scalar product, the following perpendicularities of axes are imposed:

1. $x_4 \perp z_3$ - The rotation about the y_i axes is constrained with this imposed perpendicularity.

2. $x_4 \perp y_3$ - The rotation about the z_i axes is constrained.

3. $y_4 \perp z_3$ - The rotation about the x_i axes is constrained.

The two body-fixed reference frames of bodies 3 and 4 are constrained to remain parallel at all times, meaning that during the motion the corresponding coordinate axes of the bodies have to remain mutually parallel. Mathematically, this stated using the vector scalar products

$$\begin{aligned}
\mathbf{\Phi}_1^{PJ} &= \mathbf{E}_4^{x^T} \mathbf{E}_3^z = 0, \\
\mathbf{\Phi}_2^{PJ} &= \mathbf{E}_4^{x^T} \mathbf{E}_3^y = 0, \\
\mathbf{\Phi}_3^{PJ} &= \mathbf{E}_4^{y^T} \mathbf{E}_3^z = 0.
\end{aligned}$$

When the relationship connecting the local and global coordinate systems, given with (4.13), is inserted in the equations given above, the constraint equations that eliminate the prismatic

joint relative rotations at the displacement level are obtained:

$$\begin{aligned}
\mathbf{\Phi}_1^{PJ} &= \left(\mathbf{R}_4\mathbf{e}^x\right)^T \mathbf{R}_3\mathbf{e}^z = 0,\\
\mathbf{\Phi}_2^{PJ} &= \left(\mathbf{R}_4\mathbf{e}^x\right)^T \mathbf{R}_3\mathbf{e}^y = 0,\\
\mathbf{\Phi}_3^{PJ} &= \left(\mathbf{R}_4\mathbf{e}^y\right)^T \mathbf{R}_3\mathbf{e}^z = 0.
\end{aligned} \tag{4.27}$$

The velocity level of equations (4.27) is of the form

$$\begin{aligned}
\dot{\mathbf{\Phi}}_1^{PJ} &= \left(\mathbf{R}_4\widetilde{\mathbf{\Omega}}_4\mathbf{e}^x\right)^T \mathbf{R}_3\mathbf{e}^z + \left(\mathbf{R}_4\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^z = 0,\\
\dot{\mathbf{\Phi}}_2^{PJ} &= \left(\mathbf{R}_4\widetilde{\mathbf{\Omega}}_4\mathbf{e}^x\right)^T \mathbf{R}_3\mathbf{e}^y + \left(\mathbf{R}_4\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^y = 0,\\
\dot{\mathbf{\Phi}}_3^{PJ} &= \left(\mathbf{R}_4\widetilde{\mathbf{\Omega}}_4\mathbf{e}^y\right)^T \mathbf{R}_3\mathbf{e}^z + \left(\mathbf{R}_4\mathbf{e}^y\right)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^z = 0.
\end{aligned} \tag{4.28}$$

By differentiating (4.27) twice with respect to time and after some mathematical manipulation the first set equations at the acceleration level is obtained in the form

$$\mathbf{\Phi}_{x,1}^{PJ}\ddot{\mathbf{x}}_{3,4} = \boldsymbol{\xi}_1^{PJ}, \tag{4.29}$$

where the constraint matrix and vector are defined as

$$\mathbf{\Phi}_{x,1}^{PJ} = \begin{bmatrix}
\mathbf{0} & \left(\mathbf{R}_4\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{e}}^z & \mathbf{0} & \left(\mathbf{R}_3\mathbf{e}^z\right)^T \mathbf{R}_4\widetilde{\mathbf{e}}^x\\
\mathbf{0} & \left(\mathbf{R}_4\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{e}}^y & \mathbf{0} & \left(\mathbf{R}_3\mathbf{e}^y\right)^T \mathbf{R}_4\widetilde{\mathbf{e}}^x\\
\mathbf{0} & \left(\mathbf{R}_4\mathbf{e}^y\right)^T \mathbf{R}_3\widetilde{\mathbf{e}}^z & \mathbf{0} & \left(\mathbf{R}_3\mathbf{e}^z\right)^T \mathbf{R}_4\widetilde{\mathbf{e}}^y
\end{bmatrix},$$

$$\boldsymbol{\xi}_1^{PJ} = \begin{bmatrix}
-\left[\left(\mathbf{R}_4\widetilde{\mathbf{\Omega}}_4\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{e}}^z + \left(\mathbf{R}_4\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\widetilde{\mathbf{e}}^z\right]\mathbf{\Omega}_3\\
\qquad -\left[\left(\mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^z\right)^T \mathbf{R}_4\widetilde{\mathbf{e}}^x + \left(\mathbf{R}_3\mathbf{e}^z\right)^T \mathbf{R}_4\widetilde{\mathbf{\Omega}}_4\widetilde{\mathbf{e}}^x\right]\mathbf{\Omega}_4\\
-\left[\left(\mathbf{R}_4\widetilde{\mathbf{\Omega}}_4\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{e}}^y + \left(\mathbf{R}_4\mathbf{e}^x\right)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\widetilde{\mathbf{e}}^y\right]\mathbf{\Omega}_3\\
\qquad -\left[\left(\mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^y\right)^T \mathbf{R}_4\widetilde{\mathbf{e}}^x + \left(\mathbf{R}_3\mathbf{e}^y\right)^T \mathbf{R}_4\widetilde{\mathbf{\Omega}}_4\widetilde{\mathbf{e}}^x\right]\mathbf{\Omega}_4\\
-\left[\left(\mathbf{R}_4\widetilde{\mathbf{\Omega}}_4\mathbf{e}^y\right)^T \mathbf{R}_3\widetilde{\mathbf{e}}^z + \left(\mathbf{R}_4\mathbf{e}^y\right)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\widetilde{\mathbf{e}}^z\right]\mathbf{\Omega}_3\\
\qquad -\left[\left(\mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^z\right)^T \mathbf{R}_4\widetilde{\mathbf{e}}^y + \left(\mathbf{R}_3\mathbf{e}^z\right)^T \mathbf{R}_4\widetilde{\mathbf{\Omega}}_4\widetilde{\mathbf{e}}^y\right]\mathbf{\Omega}_4
\end{bmatrix}.$$

As the rotations have been constrained, the translations still remain unconstrained, where only two of them have to be constrained. From figure 4.12 it is seen that the only allowed translation is along the z_i axes, while translations in the x_i and y_i directions are eliminated. Mathematically, this constraint is formulated by introducing the vector that connects the CoGs of bodies 3 and 4 and which is defined in the global coordinate system. This vector

has to be coincident with the joint translation axis. In order for the slider to slide along the z_3 axis, this vector must have the same direction as the unit vector along the z_3 axis. Two constraint equations are formulated (two relative DOFs are eliminated) giving the projections of the translation axis on the x_3 and y_3 axes. These projections are set to be equal to zero. Again, the vector scalar product is used:

$$\begin{aligned}\mathbf{\Phi}_4^{PJ} &= (\mathbf{x}_4 - \mathbf{x}_3)^T \mathbf{E}_3^x = 0,\\ \mathbf{\Phi}_5^{PJ} &= (\mathbf{x}_4 - \mathbf{x}_3)^T \mathbf{E}_3^y = 0.\end{aligned}$$

When relations (4.13) are introduced equations (4.30) follow:

$$\begin{aligned}\mathbf{\Phi}_4^{PJ} &= (\mathbf{x}_4 - \mathbf{x}_3)^T \mathbf{R}_3\mathbf{e}^x,\\ \mathbf{\Phi}_5^{PJ} &= (\mathbf{x}_4 - \mathbf{x}_3)^T \mathbf{R}_3\mathbf{e}^y.\end{aligned} \tag{4.30}$$

The velocity level of equations (4.30) is obtained by differentiating once with respect to time and equations (4.31) are obtained:

$$\begin{aligned}\dot{\mathbf{\Phi}}_4^{PJ} &= (\dot{\mathbf{x}}_4 - \dot{\mathbf{x}}_3)^T \mathbf{R}_3\mathbf{e}^x + (\mathbf{x}_4 - \mathbf{x}_3)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^x = 0,\\ \dot{\mathbf{\Phi}}_5^{PJ} &= (\dot{\mathbf{x}}_4 - \dot{\mathbf{x}}_3)^T \mathbf{R}_3\mathbf{e}^y + (\mathbf{x}_4 - \mathbf{x}_3)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^y = 0.\end{aligned} \tag{4.31}$$

Again, the acceleration level constraint equations are obtained by differentiating (4.30) twice with respect to time. The last two constraint equations at the acceleration level are obtained in the form

$$\mathbf{\Phi}_{x,2}^{PJ}\ddot{\mathbf{x}}_{3,4} = \boldsymbol{\xi}_2^{PJ}, \tag{4.32}$$

where the matrix $\mathbf{\Phi}_{x,2}^{PJ}$ and vector $\boldsymbol{\xi}_2^{PJ}$ are defined as

$$\begin{aligned}\mathbf{\Phi}_{x,2}^{PJ} &= \begin{bmatrix}(\mathbf{R}_3\mathbf{e}^x)^T & (\mathbf{x}_4 - \mathbf{x}_3)^T \mathbf{R}_3\widetilde{\mathbf{e}}^x & -(\mathbf{R}_3\mathbf{e}^x)^T & \mathbf{0}\\ (\mathbf{R}_3\mathbf{e}^y)^T & (\mathbf{x}_4 - \mathbf{x}_3)^T \mathbf{R}_3\widetilde{\mathbf{e}}^y & -(\mathbf{R}_3\mathbf{e}^y)^T & \mathbf{0}\end{bmatrix},\\ \boldsymbol{\xi}_2^{PJ} &= \begin{bmatrix}2\,(\dot{\mathbf{x}}_4 - \dot{\mathbf{x}}_3)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^x - (\mathbf{x}_4 - \mathbf{x}_3)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\widetilde{\mathbf{e}}^x\mathbf{\Omega}_3\\ 2\,(\dot{\mathbf{x}}_4 - \dot{\mathbf{x}}_3)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\mathbf{e}^y - (\mathbf{x}_4 - \mathbf{x}_3)^T \mathbf{R}_3\widetilde{\mathbf{\Omega}}_3\widetilde{\mathbf{e}}^y\mathbf{\Omega}_3\end{bmatrix}.\end{aligned}$$

The dimension of the null matrices $\mathbf{0}$ is $\dim(\mathbf{0}) = 1 \times 3$.

The complete matrix constraint equation of the prismatic joint is obtained by combining equations (4.29) and (4.32), so that equation (4.33) follows:

$$\mathbf{\Phi}_x^{PJ}\ddot{\mathbf{x}}_{3,4} = \boldsymbol{\xi}^{PJ}, \tag{4.33}$$

where the prismatic joint constraint matrix and vector are

$$\boldsymbol{\Phi}_x^{PJ} = \begin{bmatrix} \boldsymbol{\Phi}_{x,1}^{PJ} \\ \boldsymbol{\Phi}_{x,2}^{PJ} \end{bmatrix}, \qquad \boldsymbol{\xi}^{PJ} = \begin{bmatrix} \boldsymbol{\xi}_1^{PJ} \\ \boldsymbol{\xi}_2^{PJ} \end{bmatrix}.$$

The complete prismatic joint formulation consists of five algebraic equations, meaning that five relative DOFs have been eliminated with the constraint.

From the definition of the prismatic joint it is seen that axes z_3 and z_4 of body 3 and 4 are aligned and they represent the translation axis of the joint. Consequently, from the definition of the translation axis above, it has to be kept in mind that the translation axis has to pass through bodies 3 and 4 CoGs. This follows from the fact that the axes z_i have the origin in the bodies CoGs and that the translation axis is calculated in the global reference frame from the bodies CoGs positions.

4.2.5 Rheonomic constraints

In contrary to scleronomic constraints, which are time independent functions (in their mathematical formulation at the displacement level), rheonomic constraints are time-dependent functions and they describe imposed motions on the system. As joint equations, the rheonomic constraint equations are also mathematically introduced as a set of equations that represent additional row entries in the global system constraint matrix. This way, it is seen that the rheonomic constraints further reduce the number of system DOFs.

The joint constraint equations have to be formulated at the acceleration level and the same holds for the rheonomic constraint equations in order to be able to put them together in the global system constraint matrix (so that the DAE system of index 1 is obtained). This is achieved so that the rheonomic constraint equations are formulated at the displacement or velocity level first, and then, by differentiation, reshaped to the acceleration level.

The general form of the rheonomic constraint equations at the acceleration level is the same as that of the joint constraint equations:

$$\boldsymbol{\Phi}_{x,i}^{RH} \ddot{\mathbf{x}}_k = \boldsymbol{\xi}_i^{RH},$$

where i is the number of the rheonomic constraint and k is the number of the body or bodies involved in the constraint.

In the case study system there are three imposed motions which are defined the sequel:

1. The main satellite body (body 1) has a constant angular velocity (given with (4.1)) which, at the displacement level, results in a linear time dependence of the constraint

equation. The constraint equation is formulated directly at the velocity level:

$$\mathbf{\Phi}_{FP}^{RH} = \mathbf{R}_1\mathbf{\Omega}_1 - \mathbf{\Omega}_1^G = \mathbf{0}, \tag{4.34}$$

so that at the acceleration level the constraint mathematical formulation is

$$\mathbf{R}_1\dot{\mathbf{\Omega}}_1 = -\mathbf{R}_1\widetilde{\mathbf{\Omega}}_1\mathbf{\Omega}_1,$$

where $\widetilde{\mathbf{\Omega}}_1\mathbf{\Omega}_1 = \mathbf{0}$ follows from the vector cross product properties. It finally follows

$$\mathbf{R}_1\dot{\mathbf{\Omega}}_1 = \mathbf{0}. \tag{4.35}$$

The constraint matrix $\mathbf{\Phi}_{x,1}^{RH}$ and vector $\boldsymbol{\xi}_1^{RH}$ are, from equation (4.35), identified as

$$\mathbf{\Phi}_{x,1}^{RH} = \begin{bmatrix}\mathbf{0}^{3\times3} & \mathbf{R}_1\end{bmatrix},$$
$$\boldsymbol{\xi}_1^{RH} = \mathbf{0}^{3\times1},$$

where the superscript on the zero matrix $\mathbf{0}$ denotes the dimension of the matrix.

2. The spherical joint has a constant angular velocity (equation (4.2)), meaning that the angular velocity of body 2 is constant. The constant angular velocity of the joint is expressed as the angular velocity $\mathbf{\Omega}_2^G$ in the inertial reference frame. This constraint is formulated at the velocity level as

$$\mathbf{\Phi}_{SJ}^{RH} = \mathbf{R}_2\mathbf{\Omega}_2 - \mathbf{\Omega}_2^G = \mathbf{0}, \tag{4.36}$$

while at the acceleration level it follows (again the vector cross product property $\widetilde{\mathbf{\Omega}}_2\mathbf{\Omega}_2 = \mathbf{0}$ is used)

$$\mathbf{R}_2\dot{\mathbf{\Omega}}_2 = \mathbf{0}. \tag{4.37}$$

The constraint matrix $\mathbf{\Phi}_{x,2}^{RH}$ and vector $\boldsymbol{\xi}_2^{RH}$ are, from equation (4.37), identified as

$$\mathbf{\Phi}_{x,2}^{RH} = \begin{bmatrix}\mathbf{0}^{3\times3} & \mathbf{R}_2\end{bmatrix},$$
$$\boldsymbol{\xi}_2^{RH} = \mathbf{0}.$$

3. The prismatic joint allows only one relative translation of the slider, whose acceleration is given with (4.4) and the initial velocity with (4.3). The slider has a constant acceleration with a non-zero initial velocity, so that the rheonomic constraint, formulated at the displacement level, has the form

$$\Phi_{PJ}^{RH} = \left(\mathbf{x}_4 - \mathbf{x}_3\right)^T \mathbf{R}_3\mathbf{e}^z - \dot{x}_{3,4}^{rel}t - \ddot{x}_{3,4}^{rel}\frac{t^2}{2} = 0. \tag{4.38}$$

The velocity level follows by differentiating once with respect to time

$$\dot{\Phi}_{PJ}^{RH} = \left(\dot{\mathbf{x}}_4 - \dot{\mathbf{x}}_3\right)^T \mathbf{R}_3 \mathbf{e}^z + \left(\mathbf{x}_4 - \mathbf{x}_3\right)^T \mathbf{R}_3 \widetilde{\mathbf{\Omega}}_3 \mathbf{e}^z - \dot{x}_{3,4}^{rel} - \ddot{x}_{3,4}^{rel} t = 0 \tag{4.39}$$

The acceleration level is obtained by further differentiating with respect to time:

$$\begin{aligned} -\left(\mathbf{R}_3 \mathbf{e}^z\right)^T \ddot{\mathbf{x}}_3 - \left(\mathbf{x}_4 - \mathbf{x}_3\right)^T \mathbf{R}_3 \widetilde{\mathbf{e}}^z \dot{\mathbf{\Omega}}_3 \\ + \left(\mathbf{R}_3 \mathbf{e}^z\right)^T \ddot{\mathbf{x}}_4 = \ddot{x}_{3,4}^{rel} + 2\left(\dot{\mathbf{x}}_4 - \dot{\mathbf{x}}_3\right)^T \mathbf{R}_3 \widetilde{\mathbf{e}}^z \mathbf{\Omega}_3 \\ + \left(\mathbf{x}_4 - \mathbf{x}_3\right)^T \mathbf{R}_3 \widetilde{\mathbf{\Omega}}_3 \widetilde{\mathbf{e}}^z \mathbf{\Omega}_3. \end{aligned} \tag{4.40}$$

The constraint matrix $\mathbf{\Phi}_{x,3}^{RH}$ and vector $\boldsymbol{\xi}_3^{RH}$ are, from equation (4.40), identified as

$$\begin{aligned} \mathbf{\Phi}_{x,3}^{RH} &= \begin{bmatrix} -\left(\mathbf{R}_3 \mathbf{e}^z\right)^T & -\left(\mathbf{x}_4 - \mathbf{x}_3\right)^T \mathbf{R}_3 \widetilde{\mathbf{e}}^z & \left(\mathbf{R}_3 \mathbf{e}^z\right)^T & \mathbf{0}^{1\times 3} \end{bmatrix}, \\ \xi_3^{RH} &= \ddot{x}_{3,4}^{rel} + 2\left(\dot{\mathbf{x}}_4 - \dot{\mathbf{x}}_3\right)^T \mathbf{R}_3 \widetilde{\mathbf{e}}^z \mathbf{\Omega}_3 + \left(\mathbf{x}_4 - \mathbf{x}_3\right)^T \mathbf{R}_3 \widetilde{\mathbf{\Omega}}_3 \widetilde{\mathbf{e}}^z \mathbf{\Omega}_3. \end{aligned}$$

Equations (4.35) to (4.40) are the rheonomic constraint equations formulated at the acceleration level. It is seen that equations (4.35) and (4.37) consist of three algebraic equations so that they eliminate three DOFs, all of which rotations. Equation (4.40) is one algebraic equation imposing the translational motion of the slider and, thus, eliminating one DOF.

4.3 System governing equations

The system governing equations are formulated in the form of equation (1.8):

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_x^T \\ \mathbf{\Phi}_x & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \boldsymbol{\xi} \end{bmatrix},$$

and, in the sequel, the formulation of matrices that are present in the governing equations is described. All the matrices have to be formulated according to the convention of the system accelerations vector $\ddot{\mathbf{q}}$ given with

$$\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\mathbf{x}}_1 & \ddot{\mathbf{\Omega}}_1 & \ddot{\mathbf{x}}_2 & \ddot{\mathbf{\Omega}}_2 & \dots & \ddot{\mathbf{x}}_N & \ddot{\mathbf{\Omega}}_N \end{bmatrix}^T,$$

where $\ddot{\mathbf{x}}_i$ are the translational acceleration of the i-th body and $\ddot{\mathbf{\Omega}}_i$ its angular acceleration. The indices over the unitary and zero matrices denote the dimension of the matrix in question. The dimensions are given so that the dimension of the global system matrices is clearly seen.

In accordance with that, the mass matrix $\mathbf{M}$ is a diagonal matrix defined with its diagonal elements as

$$\operatorname{diag}(\mathbf{M}) = \begin{bmatrix} m_1\mathbf{I}^{3\times3} & \mathbf{J}_1 & m_2\mathbf{I}^{3\times3} & \mathbf{J}_2 & m_3\mathbf{I}^{3\times3} & \mathbf{J}_3 & m_4\mathbf{I}^{3\times3} & \mathbf{J}_4 \end{bmatrix}.$$

The global constraint matrix is formed out of all joints and rheonomic constraint matrices as

$$\mathbf{\Phi}_x = \begin{bmatrix} \mathbf{\Phi}_x^{FP} & \mathbf{0}^{3\times6} & \mathbf{0}^{3\times6} & \mathbf{0}^{3\times6} \\ & \mathbf{\Phi}_x^{SJ} & \mathbf{0}^{3\times6} & \mathbf{0}^{3\times6} \\ \mathbf{0}^{5\times6} & & \mathbf{\Phi}_x^{RJ} & \mathbf{0}^{5\times6} \\ \mathbf{0}^{5\times6} & \mathbf{0}^{5\times6} & & \mathbf{\Phi}_x^{PJ} \\ \mathbf{\Phi}_{x,1}^{RH} & \mathbf{0}^{3\times6} & \mathbf{0}^{3\times6} & \mathbf{0}^{3\times6} \\ \mathbf{0}^{3\times6} & \mathbf{\Phi}_{x,2}^{RH} & \mathbf{0}^{3\times6} & \mathbf{0}^{3\times6} \\ \mathbf{0}^{1\times6} & \mathbf{0}^{1\times6} & & \mathbf{\Phi}_{x,3}^{RH} \end{bmatrix}, \tag{4.41}$$

while the right-hand side global system constraint vector $\boldsymbol{\xi}$ is formed as

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}^{FP} & \boldsymbol{\xi}^{SJ} & \boldsymbol{\xi}^{RJ} & \boldsymbol{\xi}^{PJ} & \boldsymbol{\xi}_1^{RH} & \boldsymbol{\xi}_2^{RH} & \xi_3^{RH} \end{bmatrix}^T.$$

The global system force vector $\mathbf{Q}$ consists of the external forces that act on the system and non-linear forces originating from non-linear velocity terms of the Euler equation (1.5). If the force vector $\mathbf{Q}$ is defined with

$$\mathbf{Q} = \mathbf{Q}_{ext} + \mathbf{Q}_{nl},$$

then the non-linear part is defined as

$$\boldsymbol{Q}_{nl} = \begin{bmatrix} \mathbf{0}^{3\times1} & -\widetilde{\mathbf{\Omega}}_1\mathbf{J}_1\mathbf{\Omega}_1 & \mathbf{0}^{3\times1} & -\widetilde{\mathbf{\Omega}}_2\mathbf{J}_2\mathbf{\Omega}_2 & \mathbf{0}^{3\times1} & -\widetilde{\mathbf{\Omega}}_3\mathbf{J}_3\mathbf{\Omega}_3 & \mathbf{0}^{3\times1} & -\widetilde{\mathbf{\Omega}}_4\mathbf{J}_4\mathbf{\Omega}_4 \end{bmatrix}^T,$$

while the external forces part follows from the external forces that act on the system defined in section 4.1.3

$$\mathbf{Q}_{ext} = \begin{bmatrix} \mathbf{0}^{3\times1} & \mathbf{0}^{3\times1} & \mathbf{0}^{3\times1} & -\mathbf{L}_3 & \mathbf{0}^{3\times1} & \mathbf{L}_3 & \mathbf{R}_4\mathbf{F}_4 & \mathbf{0}^{3\times1} \end{bmatrix}^T.$$

Every column (or row, depending of the vector/matrix in question) corresponds to one body set of coordinates (rotational or angular set of accelerations) and, consequently, the number of columns in the global constraint matrix is equal to the number of rows in the global system force vector, which is equal to the number of system accelerations: $6N$. Also, from the very structure of the global system constraint matrix, the topology of the system can be

deduced. The first row of expression (4.41) defines the fixed point constraint which affects only the first body, all other row elements, that multiply with other bodies accelerations, are zero matrices. The spherical joint connects bodies 1 and 2, no other bodies accelerations are related in the second row. The rest is deduced in the same way as above.

Similarly, the rheonomic constraints can be interpreted and deduced on which joints and bodies the motions are imposed. Furthermore, by counting the entries in the system matrices the final number of the system DOFs (n) is established. It's seen that the global system constraints matrix has 23 rows, $K_{1,3} = 23$, for motion cases 1 and 3. The system is a spatial four body system, so that the full number of system coordinates is $6N = 24$. Using the equation for spatial systems (given in section 1.1.2), the number of system dynamic DOFs follows as

$$n_{1,3} = 6N - K_{1,3} = 24 - 23 = 1,$$

and that's the DOF in the revolute joint, whose motion follows from the forces that act on the bodies. The global system constraint matrix of the system motion case 2 has 22 rows because the rheonomic constraint of the prismatic joint (last row of (4.41)) is not included. For that case the number of dynamic DOFs is equal to

$$n_2 == 6N - K_2 = 24 - 22 = 2.$$

The dynamic DOFs of motion case 2 are the DOF in the revolute joint and the DOF in the prismatic joint. Those motions are obtained from the system governing equations integration procedure.

For the stabilisation algorithm it is required to formulate the global system constraint matrices at the displacement and velocity levels. The global system constraints vector at the displacement level is obtained by gathering equations (4.14), (4.17), (4.20), (4.23), (4.27), (4.30) and the slider relative displacement rheonomic constraint (4.38), so that the vector $\mathbf{\Phi}$ is

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}^{FP} & \mathbf{\Phi}^{SJ} & \mathbf{\Phi}^{RJ} & \mathbf{\Phi}^{PJ} & \Phi^{RH}_{PJ} \end{bmatrix}^T. \tag{4.42}$$

Similarly, the global system constraints vector $\dot{\mathbf{\Phi}}$ at the velocity level is obtained by gathering equations (4.15), (4.18), (4.21), (4.24), (4.28), (4.31) and the rheonomic constraint equations at the velocity level of equations (4.34), (4.36) and (4.39):

$$\dot{\mathbf{\Phi}} = \begin{bmatrix} \dot{\mathbf{\Phi}}^{FP} & \dot{\mathbf{\Phi}}^{SJ} & \dot{\mathbf{\Phi}}^{RJ} & \dot{\mathbf{\Phi}}^{PJ} & \mathbf{\Phi}^{RH}_{FP} & \mathbf{\Phi}^{RH}_{SJ} & \dot{\Phi}^{RH}_{PJ} \end{bmatrix}^T. \tag{4.43}$$

The rheonomic constraint equations directly formulated at the velocity level are not included

in the displacement level formulation of constraints. Their displacement level is stabilised as soon as velocities are stabilised.

In the next section the *MATLAB* implementation of the solver with the system definition is shortly presented.

4.4 Algorithm implementation

In order to obtain results using the method presented in the thesis, the solution procedure has to be implemented on a computer using a programming language or in a computer algebra system. A specially suitable software (which uses its own programming language) for technical computations is *MATLAB*. Hence, the algorithms and methods described in the thesis are implemented in *MATLAB*, where the solution of the defined system is obtained and post-processed. Simultaneously to the implementation of the methods in *MATLAB*, the multibody system is modelled in *ADAMS* whose solution serves for results verification.

All the solutions and simulations are performed on the same computer: a Dell Inspiron N5110 laptop with an Intel CORE i7 2.20 GHz processor, 8 GB of RAM, a 7200 rpm SATA hard drive and with a Windows 7 Ultimate 64-bit operating system installed. The *MATLAB* version installed is the 64 bit R2010a (7.10.0.499) version, while the *ADAMS* version is *MD ADAMS Student Edition 2011*.

4.4.1 *MATLAB* implementation

In the case study presented, the computational implementation and the solution algorithm are closely related as the programming was done specifically for the problem at hand. Because of that, the definition of the system and the algorithm implementation in *MATLAB* are presented simultaneously. The main *MATLAB* script and function files are shortly described in the sequel. From that description it will be clear how the system is defined in the program. It has to be kept in mind that the system definition consists of bodies inertial and geometrical properties, connections (joints) between bodies, imposed motions, forces and initial conditions.

sys_def.m In this script file the bodies and system constant parameters are defined. Bodies dimensions, masses and joints positions in the body-fixed reference frames are given. The inertia matrices are also calculated and constructed. The global mass matrix is constructed and the unit vectors of the coordinate system axes are defined for later

use. The system initial configuration is also defined, initial rotation matrices follow from the Euler 313 rotation sequence angles. Finally, the imposed motions parameters are also given (a switch for the slider motion by which the motion can be disabled is also included) together with the system forces.

con_def.m This is a function file that returns the system global constraint matrix, given with (4.41), and the right-hand side constraint vector $\boldsymbol{\xi}$ for a given input consisting of the system translational displacements, rotation matrices, translational and angular velocities:

$$\begin{bmatrix} \boldsymbol{\Phi}_x & \boldsymbol{\xi} \end{bmatrix} = \text{con\_def}\left(\mathbf{x}, \mathbf{R}, \dot{\mathbf{x}}, \boldsymbol{\Omega}\right).$$

con_vel_opt.m Similarly to the *sys_def.m* function file, this function returns the value of the system global constraint vector at the velocity level given in the equation (3.7). From that equation it follows that this vector equals to zero and this function is used in the stabilisation of system velocities. The function returns the vector $\dot{\boldsymbol{\Phi}}$ for the input consisting of translational and angular velocities (in the programmed implementation the translational displacements and rotation matrices are available to the function as global variables):

$$\begin{aligned} &\textbf{global x R}, \\ &\dot{\boldsymbol{\Phi}} = \text{con\_vel\_opt}\left(\dot{\mathbf{x}}, \boldsymbol{\Omega}\right). \end{aligned}$$

The function is also used when solving the system for the initial velocities.

con_disp_opt.m Returns the displacement level system global constraint vector of equation (3.6) which has to be equal to zero. This function is used for the stabilisation of displacements. The input in the function consists of the system translational displacements and rotation matrices:

$$\boldsymbol{\Phi} = \text{con\_disp\_opt}\left(\mathbf{x}, \mathbf{R}\right).$$

gov_eqn.m This function, for the given input consisting of translational displacements, rotation matrices, translational and angular velocities and the simulation time t_i, returns the vector containing the system accelerations and Lagrange multipliers calculated from (1.8). In this function system forces are introduced into the formulation and, if they are time dependent, evaluated. The function call is

$$\mathbf{F} = \text{gov\_eqn}\left(\mathbf{x}, \mathbf{R}, \dot{\mathbf{x}}, \boldsymbol{\Omega}\right).$$

init_vel.m In this function the least squares problem, using the global constraint equations at the velocity level, is numerically solved for the system initial velocities. The routine *lsqnonlin* is used for the least squares problem solution. The function input are global variables: initial rotation matrices and initial translational displacements:

$$\begin{gathered}\textbf{global x R},\\ \begin{bmatrix}\dot{\mathbf{x}}_0 & \boldsymbol{\Omega}_0\end{bmatrix} = \text{init\_vel}\,().\end{gathered}$$

stabilisation.m This function performs the stabilisation algorithm as described in section 3.4. For the given input, in the form of the integrator output, the function solves the least squares problems using the routine *lsqnonlin*, and returns the stabilised values of the system displacements and velocities:

$$\begin{bmatrix}\mathbf{x} & \mathbf{R} & \dot{\mathbf{x}} & \boldsymbol{\Omega}\end{bmatrix} = \text{stabilisation}\left(\widehat{\mathbf{x}}, \widehat{\mathbf{R}}, \widehat{\dot{\mathbf{x}}}, \widehat{\boldsymbol{\Omega}}\right).$$

Other function files include the calculation of the rotation matrices from the Euler 313 rotation angles (*rot313_matrix.m*, for the initial calculation of rotation matrices), and the *LC_perm.m* and *LC_perm_inv.m* functions for calculating the skew-symmetric matrix of a vector (using equation 2.4) and its inverse for obtaining the vector out of the matrix. Function files *exp_map.m*, *cay_map.m*, *dexpinv.m* and *dcayinv.m* serve for calculating the respective mappings.

The main solver scripts are the *euler_int.m*, *euler_int_stab.m*, *rk_mk_int.m*, *rk_mk_int_stab.m*, and the same scripts with the Cayley map instead of the exponential map have the suffix **_caym.m*. In those scripts the integration algorithm is implemented and the system integrated.

Finally, the computation time measurement procedure of the scripts is presented. Every script has been run five times consequently, so that for one script five different execution times (using *MATLAB tic* and *toc* statements) were obtained. The mean value of those times was taken as the relevant computation time.

4.4.2 *ADAMS* model

Using the system description and parameters definition of section 4.1, the *ADAMS* model was introduced as an *ADAMS/Solver* dataset, i.e. a script containing statements and function expressions that describe the system at hand. The dataset includes specifications of mass and inertia properties, geometry, body connections, imposed motions and forces in the

system. Statements about the analysis control in *ADAMS* and the output specification [7] are also included along with geometrical data for graphics output in the dataset. A motion animation is also obtained as a analysis result.

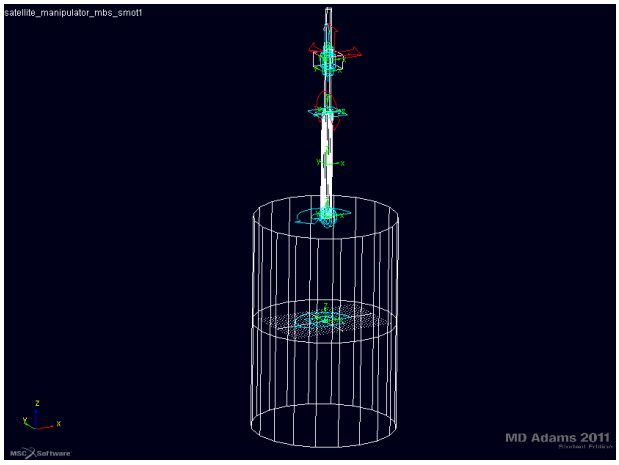


Figure 4.13: The *ADAMS* model of the MBS shown in *ADAMS/View*.

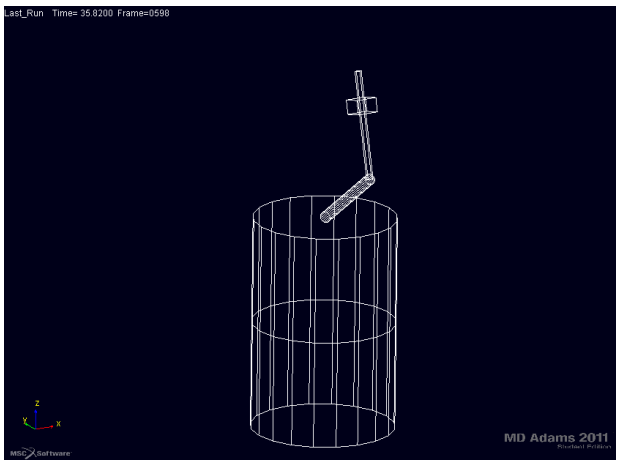


Figure 4.14: The *ADAMS* model of the MBS shown in motion.

On figure 4.13 the *ADAMS* model is shown when the *ADAMS/Solver* dataset is imported into *ADAMS/View* for visualization and simulation. Also, the system configuration obtained in *ADAMS* is presented at a simulation time t on figure 4.14.

Two different datasets are written: one for the motion case 1 and the other for the motion case 2 (motion cases are described in section 4.5). While modelling the system in *ADAMS*, the internal joints definitions of the software have to be kept in mind in order to align the local joint coordinate systems in such a manner that the software is able to connect all the bodies correctly. One such example is that for the revolute joint: the local axes z_i at the joint location of the bodies have to be aligned and with the same orientation, while in the thesis the revolute joint was formulated with the local x_i axes aligned (section 4.2.3).

4.5 Results

The case study system is solved for three different motion cases differing in the combinations of motion imposed on the bodies and in the forces acting upon the bodies. In section 4.1.3 the motion cases parameters and formulation is presented.

The simulation time for each motion case is

$$T_{sim} = T_{end} = 60\,\text{s},$$

during which the satellite manipulator performed the prescribed action or the fault was detected and the operation stopped. The results of the numerical method (of chapter 3) are obtained for the fixed integration time step length

$$h = 0.1\,\text{s},$$

with the exception of one simulation pass of motion case 1, that shows the effect of the time step, and the analysis of motion case 3.

The results of the motion cases described above are presented in the sequel. Solutions of motion cases 1 and 2 are also obtained by solving the *ADAMS* model for control purposes. The *ADAMS* solutions are then presented together with the solutions obtained using the integration method presented in the thesis.

4.5.1 Motion case 1

As it is mentioned above, the motion case 1 solution was obtained both from the *ADAMS* model (a reference solution for control) and using the method presented in the thesis. In order to show that the method presented gives correct results, figure 4.15 shows the slider rod (body 3) motion from both solutions.

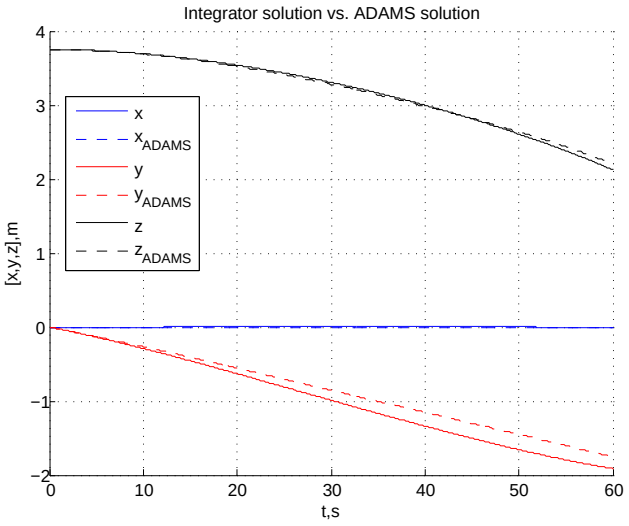


Figure 4.15: Motion 1: MK vs. *ADAMS* solution of body 3 motion for $h = 0.1$ s.

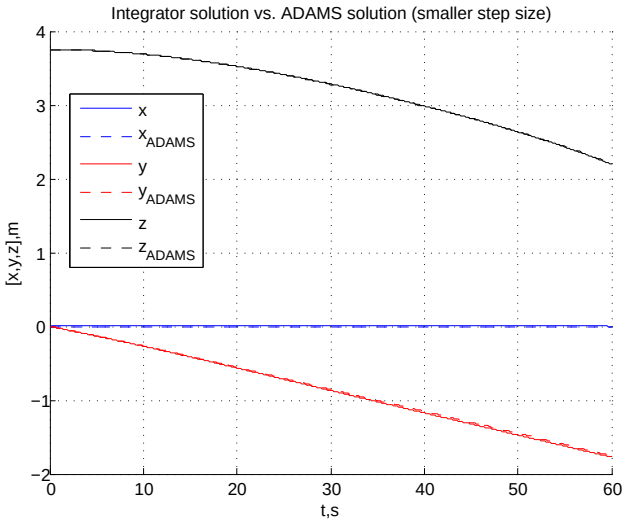


Figure 4.16: Motion 1: MK vs. *ADAMS* solution of body 3 motion for $h = 0.01$ s.

The results are shown for the body 3 as its motion is a dynamic DOF: in this way it is shown that the method presented gives correct results for the system dynamics beside the

results for DOFs that are controlled kinematically. On figure 4.15 some result discrepancies are noticed. They follow from the difference in the numerical algorithms of the MK method and the *ADAMS* solver: in the MK method a fixed time step is used while *ADAMS* uses an integration routine with a variable time step (the solution output time step was chosen to be the same). When a smaller time step length ($h = 0.01$ s) is used within the presented method, an excellent coincidence between results, seen from figure 4.16, is achieved.

Based on the presented results, it is visible that integration results of the presented method converge to the *ADAMS* solution when the integration time step is refined. It is important to mention that the solutions given above are obtained when the stabilisation algorithm is implemented within the solution algorithm. Figure 4.17 shows the solutions of the MK method with and without stabilisation.

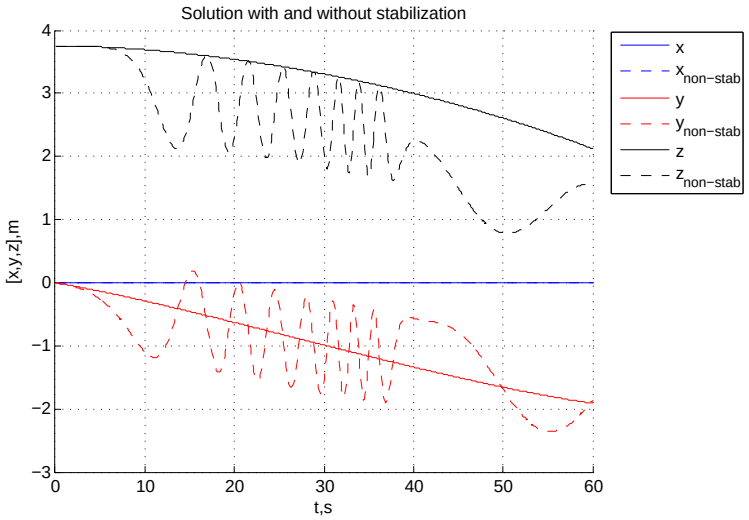


Figure 4.17: Motion 1: MK solution of body 3 motion with and without stabilisation.

When the results stabilisation (in every integration step) is not performed the wrong solution is obtained, i.e. the solution oscillates and the dynamics of the system is different. In the sequel only solutions obtained using the stabilised integration methods are presented.

In the thesis both the Lie group Euler and Munthe-Kaas integration algorithm for integrating on $SO(3)$ are presented (though the focus is mainly on the MK method) and the difference between their solution is shown on figure 4.18. It is seen that the difference between the two solutions is small (10^{-3}) compared to the integration time step (10^{-1}). The system under consideration in the case study is an open kinematic chain system and the Lie group Euler method gives solutions comparable to the more sophisticated MK method.

In order to present that the solver gives correct results for coordinates with imposed

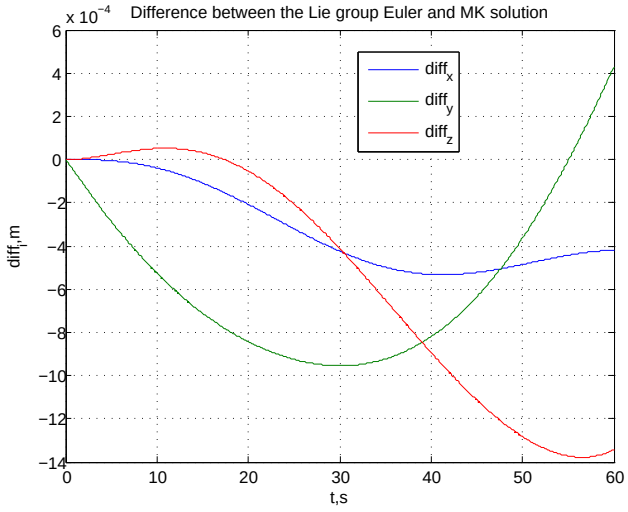


Figure 4.18: Motion 1: Difference between the MK and Lie group Euler methods solution of body 3 motion.

motions, the relative motion of the slider (body 4) expressed in the slider rod (body 3) coordinate system is presented on figure 4.19. It is seen that the curve has the shape of the time function given in equation (4.38).

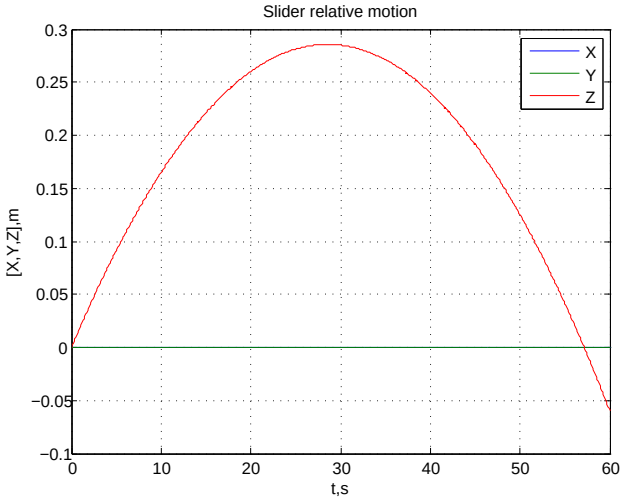


Figure 4.19: Motion 1: Slider motion seen in the slider rod CS.

Finally, all four bodies rotations are presented with a set of four plots given in figure 4.20, where in each plot the nine components of the respective body rotation matrix are plotted against time.

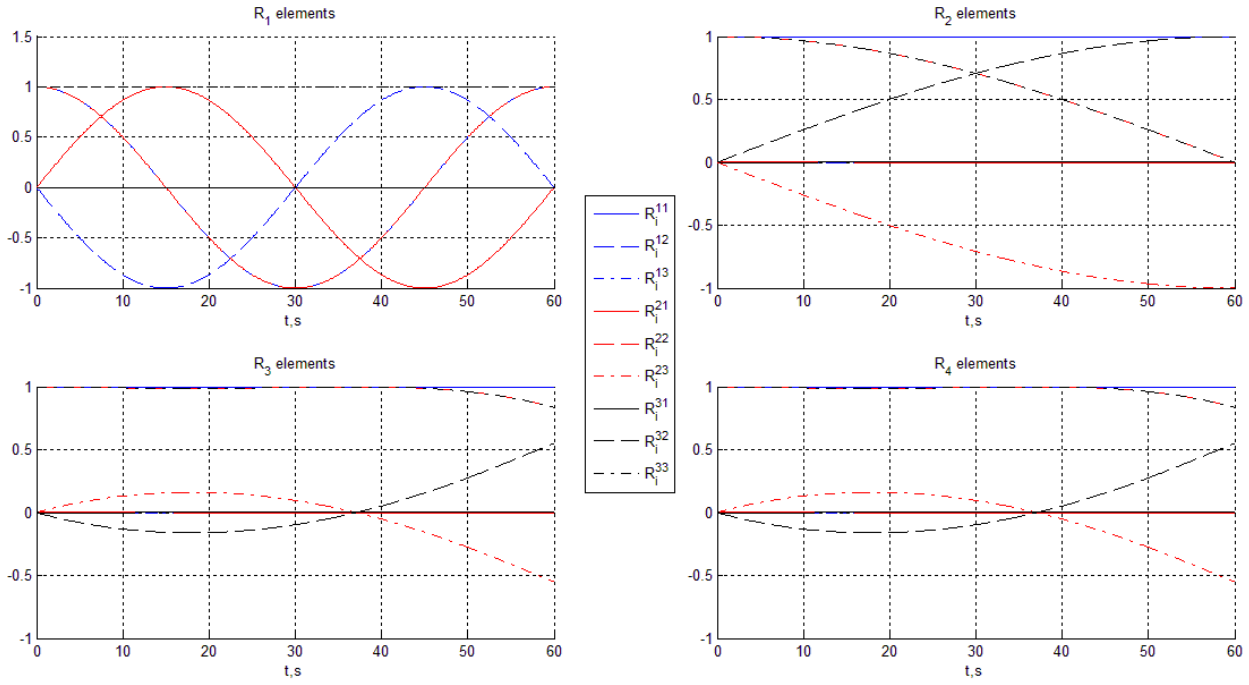


Figure 4.20: Motion 1: Elements of bodies rotation matrices.

4.5.2 Motion case 2

Same as motion case 1, the MK method solution of motion case 2 is compared to the *ADAMS* solution of the system. The difference between the motion case 1 is that now the system has two dynamic DOFs whose motion is not known a priori (i.e. for body 3 it is controlled, but dynamically). To find this motions the system has to be integrated. On figures 4.21 and 4.22 the motion of bodies 3 and 4 (slider rod and slider), obtained with the MK method, is shown in terms of the global coordinate system. On the same plots the *ADAMS* solution is presented with dashed lines.

The MK and *ADAMS* solutions coincide exactly, though the used time step length is the larger, $h = 0.1\,\mathrm{s}$, step. This results coincidence is due to the fact that, in motion case 1, one of the imposed motions is the slider translational motion with respect to the slider motion. This motion mathematically manifests as one additional equation which has to be stabilised at each integration step. When a longer time step is used with that imposed motion, together with the combination of the larger force that acts on the slider in motion case 1, the solver is less accurate. The equations have to ”transfer” the force from the slider to the slider rod at each step and in order to obtain more precise results a shorter time step has to be used in the motion case 1. In the motion case 2 this is not the case and the slider force is small and

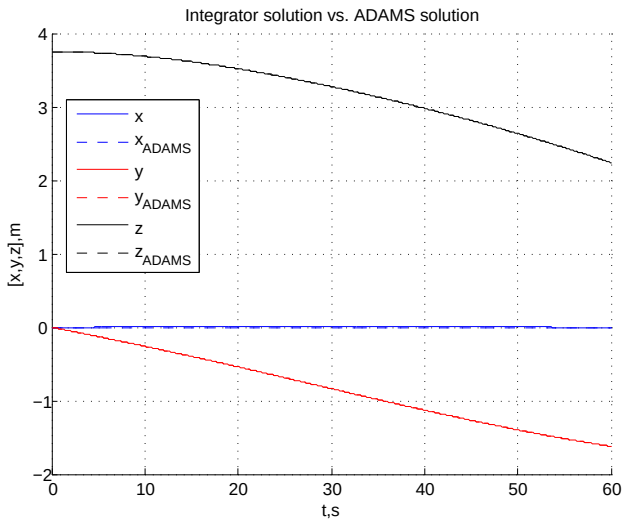


Figure 4.21: Motion 2: Body 3 motion (MK and *ADAMS*).

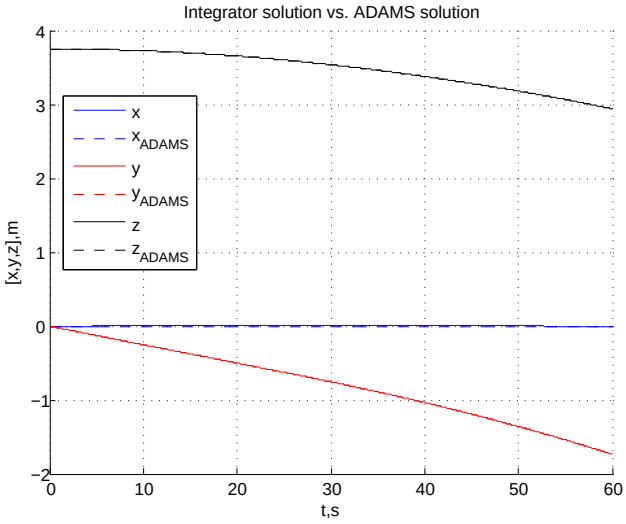


Figure 4.22: Motion 2: Body 4 motion (MK and *ADAMS*).

collinear with the slider translation axis. This results in a smaller gradient in the equations and, together with the smaller set of equations, a precise solution, even with the larger time step, is obtained.

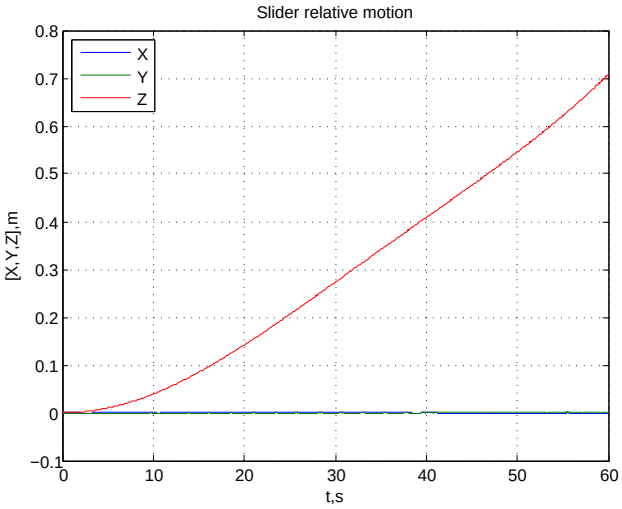


Figure 4.23: Motion 2: Slider motion seen in the slider rod CS.

On figure 4.23 the slider relative motion, seen in the slider rod coordinate system, is presented. Knowing the fact that on the slider acts a constant force directed along its translation axis, one could expect its relative motion to be a quadratic function, but it is not so. The resulting curve is not a purely quadratic function because of the centrifugal

force that acts on the slider: the centrifugal forces are variable because the slider changes its position on the slider rod, causing a change in the centripetal acceleration of the slider.

4.5.3 Motion case 3

Results of the motion case 3 are obtained using the MK method with the implemented stabilisation algorithm and a fixed time step length of

$$h = 0.01 \text{ s},$$

because of the higher angular velocity of body 1 imposed in the system. The higher velocity implies that a time step which yields good results for a slower system does not yield the same quality of results for a faster system. Also, the accuracy of the solution of motion case 3 is affected by the fact that the slider exerts an imposed motion relative to the slider rod, that influence is explained in section 4.5.2 above.

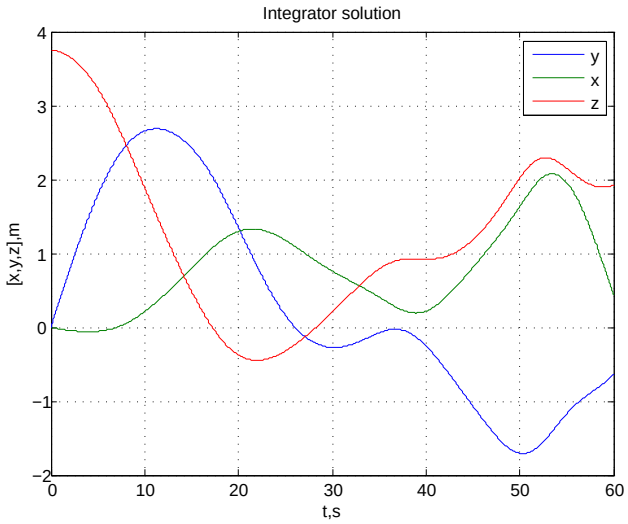


Figure 4.24: Motion 3: Body 3 (slider rod) motion (MK method).

On figure 4.24 the motion of the slider rod CoG in the inertial reference frame is shown. Also, rotation matrices of the system bodies are presented on figure 4.25, where each body rotation matrix is represented with its nine components in one plot. From the results of figure 4.25 it is seen that the rotations are large. Furthermore, it is seen that bodies 3 and 4 have the same orientation. That could have been anticipated since these bodies are connected with the prismatic joint that eliminates relative rotations.

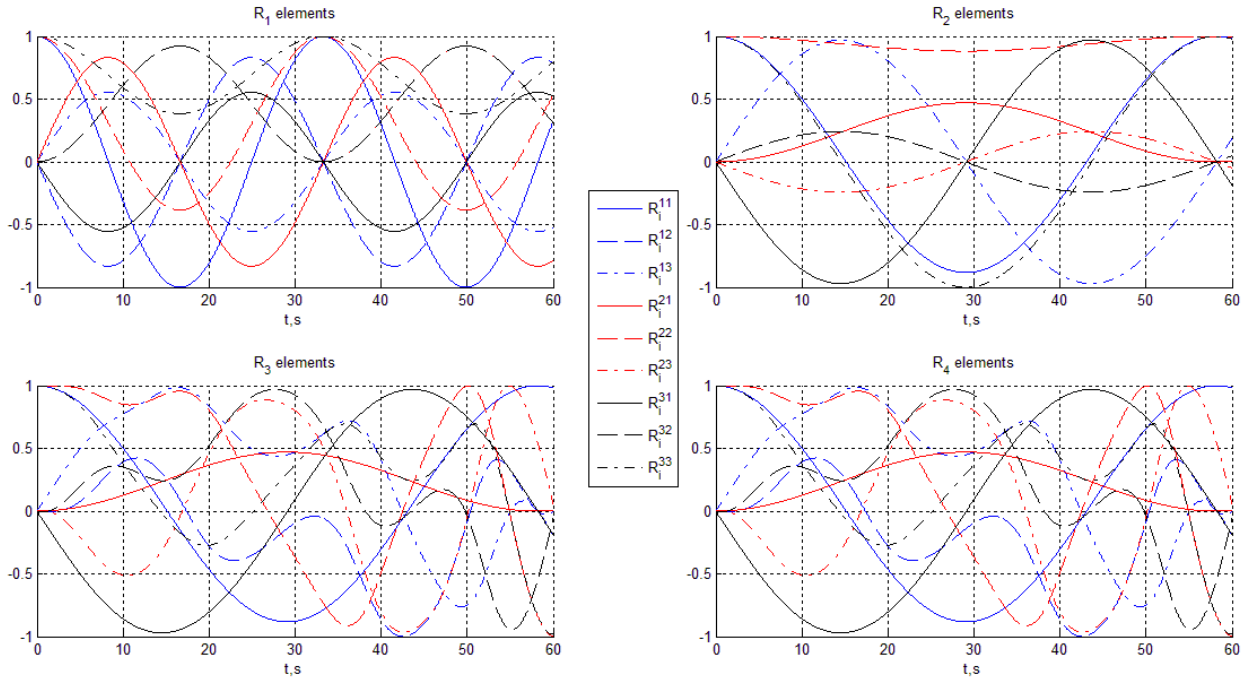


Figure 4.25: Motion 3: Elements of bodies rotation matrices.

An additional results verification is the check of the rotation matrices properties. Rotation matrices are elements of $SO(3)$ (defined with the relation (2.2)), which are orthogonal matrices with a positive determinant equal to one. For every simulation time (for which the solution is calculated) this properties of bodies rotation matrices are checked. Results are shown on figure 4.26.

Also, this verification, besides proving that the solution remains on $SO(3)$, shows that the stabilisation algorithm (step 2 of the algorithm given in section 3.4) works correctly and its solution satisfies the rotation matrices orthogonality condition.

4.5.4 Exponential and Cayley map comparison

The Cayley map was shortly presented in section 3.5.2, where the relation (3.12) for its practical computation was given. Compared to the exponential map of equation (3.9), it is seen that the Cayley map is, theoretically, less computationally expensive. This is the main advantage of the Cayley map, but the existence of singularities in the map are its main drawback and limit the practical application of the map.

In table 4.5.4 the computation times of different system integration procedures are presented for both the exponential and Cayley map application. From the results it is seen

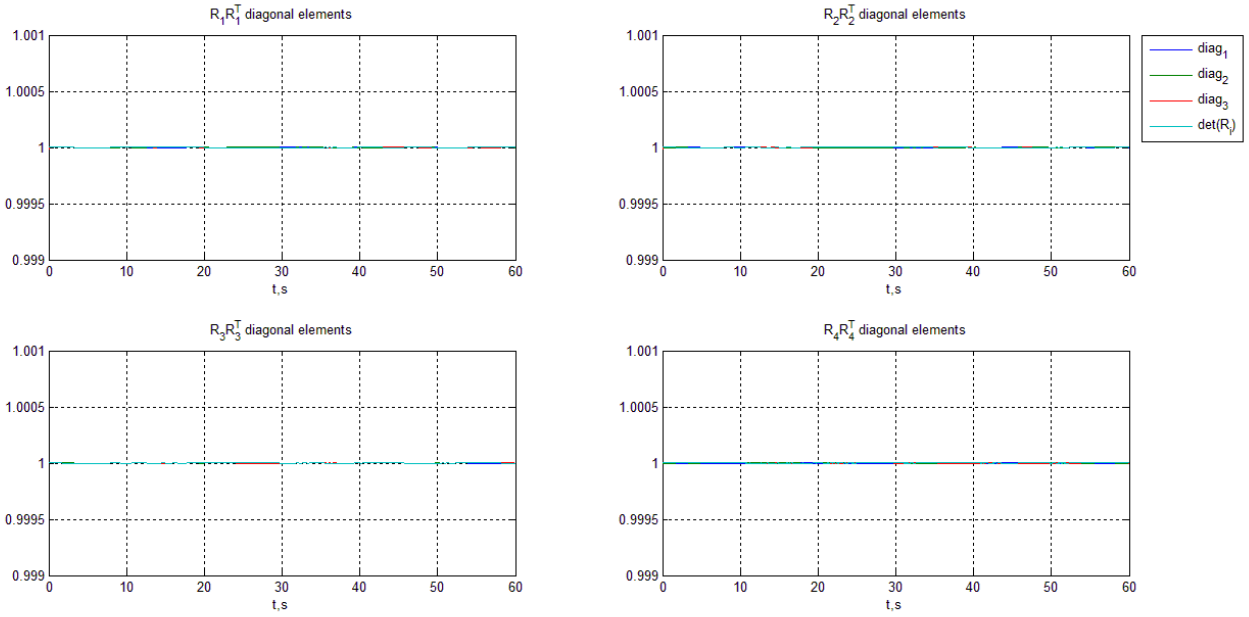


Figure 4.26: Motion 3: Properties of the rotation matrices $\mathbf{R}_i \in SO\,(3)$.

that the difference is very small and the Cayley map is faster than the exponential one only when the Euler method is applied. When the system is solved using the MK method the computation times are virtually the same and marginally faster in the case of the exponential map. If the stabilisation algorithm is applied, the computation time is extended significantly by the stabilisation algorithm. The influence of the stabilisation is then much greater than the influence of the map used.

	Without stabilisation		**With stabilisation**	
	T_{exp}, s	T_{cay}, s	T_{exp}, s	T_{cay}, s
Euler	1.119	1.109	28.775	28.452
RK-MK	3.728	3.729	26.516	26.556

Table 4.1: Solution algorithm computation times for the exponential and Cayley maps.

As the stabilisation has to be implemented in order to obtain correct results, the focus is laid upon those computation times. It is seen that the MK method is, in the stabilised case, faster than the Lie group Euler in the context of the presented case. This is due to the fact that in each integration time step the MK solution is more precise than the Lie group Euler solution. A more precise intermediate integrator solution makes the stabilisation algorithm work faster and, consequently, the correct solution of the integration step is obtained faster.

Chapter 5

Conclusion

In this chapter all partial conclusions made in the thesis are summarized and presented. The conclusions are backed up with the case study results. The method's advantages are pointed out clearly.

In chapter 1, it was shown that the dynamics of mechanical systems consisting of more bodies can be mathematically modelled as a system of differential algebraic equations (DAE system) which include informations about the system inertia and the bodies connections. By solving equation (1.8) or (1.7), the motion of the system is obtained and other quantities, like system forces, forces on a specific element, loads on joints and other data that is required for system design and strength calculations, can be obtained.

In chapter 2 it was shown that rotations of three-dimensional bodies form a group of special orthogonal matrices denoted with $SO(3)$. The special properties of this group, together with the geometrical interpretation of the group ($SO(3)$ is a manifold with elements that posses group properties, i.e. it is a Lie group), provide the basis for developing integration methods that integrate the body rotational motion directly on the Lie group, while respecting the group structure and properties at the same time. In order to be able to integrate on the group, the differential equations have to be formulated on the Lie group and the general solution form obtained. It was shown that the exponential map is the solution of the initial value problem presented and solved in section 2.3.4. Thus, the exponential map provides mathematical means used for developing integration methods on the manifold.

After the integration methods presentation, a case study was performed to show the advantages of the method. The Lie group integration methods presented in the thesis yield correct results, where the reference results (taken to be correct) are the *ADAMS* results. The effect of the time step length used in the integration methods is seen from the results (see

section 4.5.1). Also, the time step influence depends upon the system formulation. Different systems (motion cases) give different result errors for the same integration time step length h in the case of Lie group methods. Motion cases 1 and 2 (see sections 4.5.1 and 4.5.2) show this influence.

Within the presented Lie group integration methods, a constraint violation stabilisation algorithm, based on solving the least squares problem, is described. The stabilisation algorithm numerically minimizes the constraint violation at the velocity and displacement level, thus returning the solution on the system configuration manifold.

One of the main advantages of Lie group formulation is the non-existence of singularities that would have occurred if the system had been formulated in vector space and integrated using a standard integration method. Therefore, motion case 3 was formulated (see section 4.5.3) to present results for the case of large rotational domains. If the local parametrizations of $SO\,(3)$ have been used, singularities would be encountered.

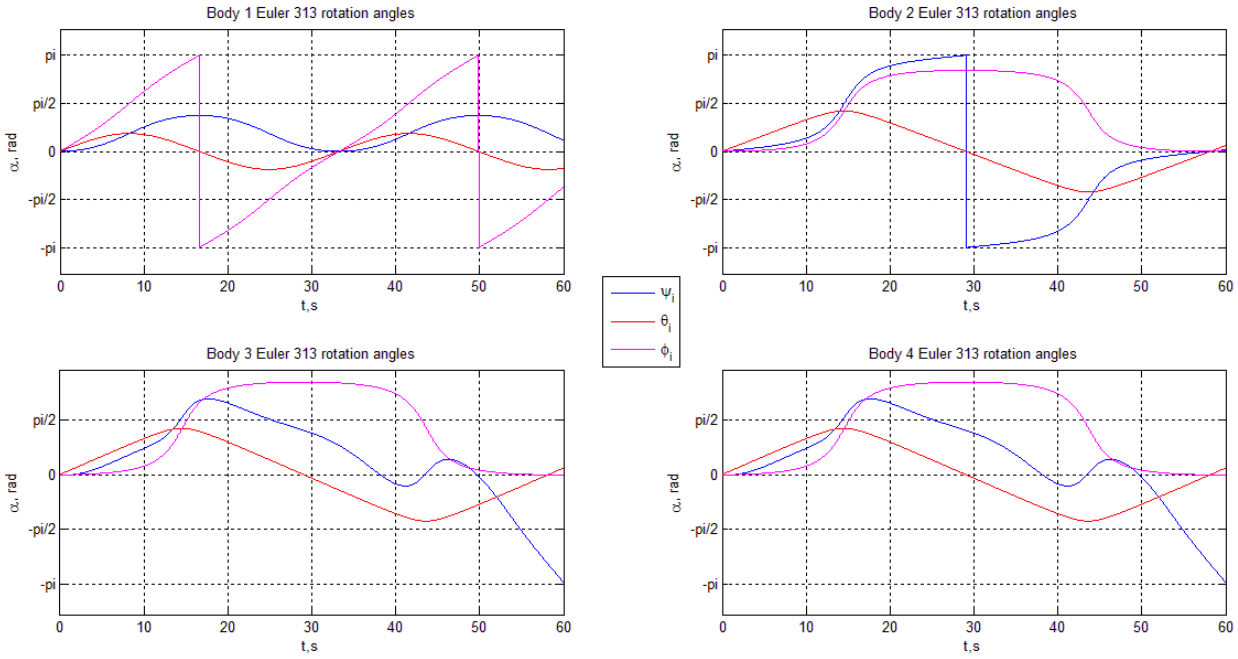


Figure 5.1: Motion 3: Euler ψ, θ and φ rotation angles of the system bodies.

On figure 5.1 the Euler angles of the system bodies are presented for the motion case 3. The angles were obtained by post-processing the rotation matrices of the bodies. The Euler rotation angles domain is $-\pi \leq \theta_i \leq \pi$. From the result plots two singularity occurrences for the body 1 motion and one occurrence for the body 2 motion are noticed. Bodies 3 and 4 motion is such that their rotations are not so large and singularities don't occur.

Since they operate directly with angular velocities and rotation matrices, the Lie group integration methods of chapter 3 are able to solve large rotational motions of the system without encountering singularities. This conclusion follows from the results presented on figure 5.1.

The Lie group integration methods circumvent problems of kinematic singularities that arise due to the use of rotation parametrizations. The need for re-parametrizations is, thus avoided. Within the presented case study, the methods showed numerical robustness and accuracy as the results are coincident with the *ADAMS* results. Also, the methods are easy to implement on general multibody problems.

References

[1] H. Baruh. *Analytical Dynamics*. WCB/McGraw-Hill, 1999.

[2] E. Celledoni and B. Owren. Lie group methods for rigid body dynamics and time integration on manifolds. *Computer methods in applied mechanics and engineering*, (192):421–438, 2003.

[3] A. Iserles, H. Z. Munthe-Kaas, S. P. Norsett, and A. Zanna. Lie-group methods. *Acta Numerica 2000*, 9:215–365, 2000.

[4] J. V. Jose and E. J. Saletan. *Classical dynamics: a contemporary approach*. Cambridge University Press, 1998.

[5] E. Kreyszig. *Advanced engineering mathematics*. John Wiley & Sons, 9th edition, 2006.

[6] A. Morawiec. *Orientations and Rotations: Computations in Crystallographic Textures*. Springer, 2004.

[7] MSC Software Corporation. *ADAMS/Solver MDR3 manual*, October 2009. Availible at http://simcompanion.mscsoftware.com.

[8] B. Schutz. *Geometrical methods of mathematical physics*. Cambridge University Press, 1980.

[9] Z. Terze and A. Eiber. Dynamics of Multibody Systems: modelling concepts and applications. http://www.fsb.unizg.hr/aero/, 2009. Internal version.

[10] Z. Terze and J. Naudet. Geometric properties of projective constraint violation stabilization method for generally constrained multibody systems on manifolds. *Multibody System Dynamics*, (20):85–106, 2008.

[11] Z. Terze, D. Zlatar, and A. Müller. Lie-group integration method for constrained multibody systems. Proceedings of ECCOMAS Thematic Conference Multibody Dynamics 2011, Brussels, Belgium, July 2011.

[12] Z. Terze, D. Zlatar, and A. Mueller. Lie-group integration method for constrained multibody systems in stabilized DAE index 1 form. *Multibody System Dynamics*, To appear.